Mayur Naik

Associate Professor
Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania

610 Levine Hall | 3330 Walnut St, Philadelphia, PA 19104
Email: mhnaik@cis.upenn.edu
Web: http://www.seas.upenn.edu/~mhnaik/
Phone: +1-215-573-1856

1. Education

Degree	Year	University	Field
Ph.D.	2008	Stanford University Stanford, CA *Dissertation:* Effective Static Race Detection for Java *Advisor:* Alexander Aiken	Computer Science
M.S.	2003	Purdue University W. Lafayette, IN *Dissertation:* A Type System Equivalent to Model Checking *Advisor:* Jens Palsberg	Computer Science
B.E.	1999	Birla Institute of Technology and Science Pilani, India	Computer Science

2. Employment History

Title	Organization	Years
Associate Professor	University of Pennsylvania	08/2016 – present
Assistant Professor	Georgia Institute of Technology	07/2011 – 08/2016
Research Scientist	Intel Research	10/2007 – 05/2011

3. Honors and Awards

3.1. National Awards

- NSF CAREER Award, 2013.
- Microsoft Software Engineering Innovation Foundation (SEIF) Award, 2012.
- Google Faculty Research Award, 2011.
- Microsoft Research Graduate Fellowship, 2004-2005.

3.2. Publication Awards

- ACM SIGSOFT Distinguished Paper Award, 2015.
  For the paper titled "A User-Guided Approach to Program Analysis" in Proceedings of the 23rd Symposium on Foundations of Software Engineering (FSE'15).
- ACM SIGPLAN Distinguished Paper Award, 2014.
  For the paper titled "On Abstraction Refinement for Program Analyses in Datalog" in Proceedings of the 35th ACM Conference on Programming Language Design and Implementation (PLDI'14).

- ACM SIGSOFT Distinguished Artifact Award, 2013.
  For the paper titled "Dynodroid: An Input Generation System for Android Apps" in Proceedings of the 21st Symposium on Foundations of Software Engineering (FSE'13).
- ACM SIGSOFT Distinguished Paper Award, 2009.
  For the paper titled "Effective Static Deadlock Detection" in Proceedings of the 31st International Conference on Software Engineering (ICSE'09).

3.3. University Awards

- Outstanding Junior Faculty Research Award, Georgia Tech, 2016.
  Awarded for "the quality of publications and the significance and impact of research."
- Lockheed-Martin Teaching Excellence Award, Georgia Tech, 2015.
  Awarded for "extraordinary effectiveness in classroom teaching, educational innovations, inspiration transmitted to students, direct impact and involvement with students, and impact on the postgraduate success of students."

4. Publications

4.1. Book Chapters

1. Jens Palsberg and Mayur Naik. ILP-based resource-aware compilation. In *Multiprocessor Systems-on-Chips*. Morgan Kaufmann, 2004.

4.2. Edited Volumes

1. Anders Moeller and Mayur Naik. Proceedings of the 4th ACM SIGPLAN International Workshop on the State Of The Art in Java Program Analysis (SOAP 2015), Portland, Oregon, USA. Co-located with PLDI'15, June 2015.

4.3. Journal Articles

1. Yongin Kwon, Byung-Gon Chun, Hayoon Yi, Donghyun Kwon, Seungjun Yang, Sangmin Lee, Ling Huang, Petros Maniatis, Mayur Naik, Yunheung Paek. Mantis: Efficient predictions of execution time, energy usage, memory usage and network usage on smart mobile devices. *IEEE Transactions on Mobile Computing*, 14(10):2059–2072, 2015.
2. David Gay, Joel Galenson, Mayur Naik, Kathy Yelick. Yada: Straightforward parallel programming. *Parallel Computing*, 37(9):592–609, 2011.
3. Mayur Naik and Jens Palsberg. A type system equivalent to a model checker. *ACM Transactions on Programming Languages and Systems*, 30(5), 2008.
4. Mayur Naik and Jens Palsberg. Compiling with code-size constraints. *ACM Transactions on Embedded Computing Systems*, 3(1):163–181, 2004.

4.4. Invited Tutorial Papers

1. Xujie Si, Xin Zhang, Radu Grigore, Mayur Naik. Maximum satisfiability in program analysis: applications and techniques. In *International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'18)*, January 2018.
2. Xujie Si, Xin Zhang, Radu Grigore, Mayur Naik. Maximum satisfiability in software analysis: applications and techniques. In *International Conference on Computer Aided Verification (CAV'17)*, June 2017.

4.5. Refereed Conference Papers

1. Mukund Raghothaman, Sulekha Kulkarni, Kihong Heo, Mayur Naik. User-guided program reasoning using Bayesian inference. In *ACM Conference on Programming Language Design and Implementation (PLDI'18)*, June 2018.
2. Woosuk Lee, Kihong Heo, Rajeev Alur, Mayur Naik. Accelerating search-based program synthesis using learned probabilistic models. In *ACM Conference on Programming Language Design and Implementation (PLDI'18)*, June 2018.
3. Xin Zhang, Radu Grigore, Xujie Si, Mayur Naik. Effective interactive resolution of static analysis alarms. In *Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'17)*, October 2017.
4. Sulekha Kulkarni, Ravi Mangal, Xin Zhang, Mayur Naik. Accelerating program analyses by cross-program training. In *Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'16)*, November 2016.
5. Xujie Si, Xin Zhang, Vasco Manquinho, Mikolas Janota, Alexey Ignatiev, Mayur Naik. On incremental core-guided MaxSAT solving. In *International Conference on Principles and Practice of Constraint Programming (CP'16)*, September 2016.
6. Insu Yun, Changwoo Min, Xujie Si, Yeongjin Jang, Taesoo Kim, Mayur Naik. APISan: Sanitizing API usages through semantic cross-checking. In *USENIX Security Symposium (Security'16)*, August 2016.
7. Ravi Mangal, Xin Zhang, Aditya Kamath, Aditya V. Nori, Mayur Naik. Scaling relational inference using proofs and refutations. In *Conference on Artificial Intelligence (AAAI'16)*, February 2016.
8. Xin Zhang, Ravi Mangal, Aditya V. Nori, Mayur Naik. Query-guided maximum satisfiability. In *ACM Symposium on Principles of Programming Languages (POPL'16)*, January 2016.
9. Ravi Mangal, Xin Zhang, Aditya V. Nori, Mayur Naik. Volt: A lazy grounding framework for solving very large MaxSAT instances. In *International Conference on Theory and Applications of Satisfiability Testing (SAT'15)*, September 2015.
10. Ghila Castelnuovo, Mayur Naik, Noam Rinetzky, Mooly Sagiv, Hongseok Yang. Modularity in lattices: A case study on the correspondence between top-down and bottom-up analysis. In *International Static Analysis Symposium (SAS'15)*, September 2015.
11. Ravi Mangal, Xin Zhang, Aditya V. Nori, Mayur Naik. A user-guided approach to program analysis. In *Symposium on Foundations of Software Engineering (FSE'15)*, August 2015.
12. Jongse Park, Hadi Esmaeilzadeh, Xin Zhang, Mayur Naik, William Harris. FLEXJAVA: Language support for safe and modular approximate programming. In *Symposium on Foundations of Software Engineering (FSE'15)*, August 2015.
13. Cong Shi, Karim Habak, Pranesh Pandurangan, Mostafa Ammar, Mayur Naik, Ellen Zegura. COSMOS: Computation offloading as a service for mobile devices. In *ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'14)*, August 2014.
14. Xin Zhang, Ravi Mangal, Mayur Naik, Radu Grigore, Hongseok Yang. On abstraction refinement for program analyses in Datalog. In *ACM Conference on Programming Language Design and Implementation (PLDI'14)*, June 2014.
15. Xin Zhang, Ravi Mangal, Mayur Naik, Hongseok Yang. Hybrid top-down and bottom-up interprocedural analysis. In *ACM Conference on Programming Language Design and Implementation (PLDI'14)*, June 2014.
16. Ravi Mangal, Mayur Naik, Hongseok Yang. A correspondence between two approaches to interprocedural analysis in the presence of join. In *European Symposium on Programming (ESOP'14)*, April 2014.
17. Aravind Machiry, Rohan Tahiliani, Mayur Naik. Dynodroid: An input generation system for Android apps. In *Symposium on Foundations of Software Engineering (FSE'13)*, August 2013.

18. Yongin Kwon, Sangmin Lee, Hayoon Yi, Donghyun Kwon, Seungjun Yang, Byung-Gon Chun, Ling Huang, Petros Maniatis, Mayur Naik, Yunheung Paek. Mantis: Automatic performance prediction for smartphone applications. In *USENIX Annual Technical Conference (ATC'13)*, June 2013.
19. Xin Zhang, Mayur Naik, Hongseok Yang. Finding optimum abstractions in parametric dataflow analysis. In *ACM Conference on Programming Language Design and Implementation (PLDI'13)*, June 2013.
20. Saswat Anand, Mayur Naik, Hongseok Yang, Mary Jean Harrold. Automated concolic testing of smartphone apps. In *Symposium on Foundations of Software Engineering (FSE'12)*, November 2012.
21. Mayur Naik, Hongseok Yang, Ghila Castelnuovo, Mooly Sagiv. Abstractions from tests. In *ACM Symposium on Principles of Programming Languages (POPL'12)*, January 2012.
22. Percy Liang and Mayur Naik. Scaling abstraction refinement via pruning. In *ACM Conference on Programming Language Design and Implementation (PLDI'11)*, June 2011.
23. Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, Ashwin Patti. CloneCloud: elastic execution between mobile device and cloud. In *European Conference on Computer Systems (EuroSys'11)*, April 2011.
24. Percy Liang, Omer Tripp, Mayur Naik. Learning minimal abstractions. In *ACM Symposium on Principles of Programming Languages (POPL'11)*, January 2011.
25. Ling Huang, Jinzhu Jia, Bin Yu, Byung-Gon Chun, Petros Maniatis, Mayur Naik. Predicting execution time of computer programs using sparse polynomial regression. In *Conference on Neural Information Processing Systems (NIPS'10)*, December 2010.
26. Pallavi Joshi, Mayur Naik, Koushik Sen, David Gay. An effective dynamic analysis for detecting generalized deadlocks. In *Symposium on Foundations of Software Engineering (FSE'10)*, November 2010.
27. Percy Liang, Omer Tripp, Mayur Naik, Mooly Sagiv. A dynamic evaluation of the precision of static heap abstractions. In *Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'10)*, October 2010.
28. Pallavi Joshi, Mayur Naik, Chang-Seo Park, Koushik Sen. Calfuzzer: an extensible active testing framework for concurrent programs. In *Conference on Computer Aided Verification (CAV'09)*, June 2009.
29. Zachary Anderson, David Gay, Mayur Naik. Lightweight annotations for controlling sharing in concurrent data structures. In *ACM Conference on Programming Language Design and Implementation (PLDI'09)*, June 2009.
30. Pallavi Joshi, Chang-Seo Park, Koushik Sen, Mayur Naik. A randomized dynamic program analysis technique for detecting real deadlocks. In *ACM Conference on Programming Language Design and Implementation (PLDI'09)*, June 2009.
31. Mayur Naik, Chang-Seo Park, Koushik Sen, David Gay. Effective static deadlock detection. In *International Conference on Software Engineering (ICSE'09)*, May 2009.
32. Mayur Naik and Alex Aiken. Conditional must not aliasing for static race detection. In *ACM Symposium on Principles of Programming Languages (POPL'07)*, January 2007.
33. Mayur Naik, Alex Aiken, John Whaley. Effective static race detection for Java. In *ACM Conference on Programming Language Design and Implementation (PLDI'06)*, June 2006.
34. Alice X. Zheng, Michael I. Jordan, Ben Liblit, Mayur Naik, Alex Aiken. Statistical debugging: simultaneous identification of multiple bugs. In *International Conference on Machine Learning (ICML'06)*, June 2006.
35. Ben Liblit, Mayur Naik, Alice X. Zheng, Alex Aiken, Michael I. Jordan. Scalable statistical bug isolation. In *ACM Conference on Programming Language Design and Implementation (PLDI'05)*, June 2005.
36. Mayur Naik and Jens Palsberg. A type system equivalent to a model checker. In *European Symposium on Programming (ESOP'05)*, April 2005.
37. Thomas Ball, Mayur Naik, Sriram K. Rajamani. From symptom to cause: localizing errors in counterexample traces. In *ACM Symposium on Principles of Programming Languages (POPL'03)*, January 2003.
38. Mayur Naik and Jens Palsberg. Compiling with code-size constraints. In *Conference on Languages, Compilers,*

and Tools for Embedded Systems (LCTES'02), June 2002.

4.6. Workshop Papers

1. Xin Zhang, Xujie Si, Mayur Naik. Combining the logical and the probabilistic in program analysis. In *ACM SIGPLAN Workshop on Machine Learning and Programming Languages (MAPL'17)*, June 2017.
2. Jongse Park, Kangqi Ni, Xin Zhang, Hadi Esmaeilzadeh, Mayur Naik. Expectation-oriented framework for automating approximate programming. In *Workshop on Approximate Computing Across the System Stack (WACAS'14)*, March 2014.
3. Cong Shi, Mostafa Ammar, Ellen Zegura, Mayur Naik. Computing in cirrus clouds: The challenge of intermittent connectivity. In *ACM SIGCOMM Mobile Cloud Computing Workshop (MCC'12)*, August 2012.
4. Ben Liblit, Alice X. Zheng Mayur Naik, Alex Aiken, Michael I. Jordan. Public deployment of cooperative bug isolation. In *Workshop on Remote Analysis and Measurement of Software Systems (RAMSS'04)*, May 2004.

5. Presentations

5.1. Invited Talks

- "Maximum Satisfiability in Software Analysis: Applications and Techniques." Invited Tutorial, International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'18), January 2018.
- "Hunting Software Bugs Using Machine Learning." Distinguished Lecture, Iowa State University, October 2017.
- "Maximum Satisfiability in Software Analysis: Applications and Techniques." Invited Tutorial, International Conference on Computer Aided Verification (CAV'17), July 2017.
- "On Abstraction Refinement for Program Analyses in Datalog." India Software Engineering Conference (ISEC'15), February 2015.
- "Self-Adaptive Static Analysis." 2nd Workshop on Software Correctness and Reliability, ETH Zurich, Switzerland, October 2014.
- "Large-Scale Configurable Static Analysis." 3rd ACM SIGPLAN International Workshop on the State Of The Art in Java Program Analysis (SOAP'14), Edinburgh, UK, June 2014.
- "Automated Testing of Mobile Apps." 1st International Workshop on Challenges in Mobile Apps: A Multi-Disciplinary Perspective, Toronto, Canada, November 2013.
- "Mechanizing Program Analysis with Chord." 1st International Workshop on Learning From Experience (LFX'10), Toronto, Canada, June 2010.
- "Effective Static Race Detection for Java." Workshop on Architectures and Compilers for Multi-threading, IIT Kanpur, India, December 2007.

5.2. Tutorials

- "Chord: A Versatile Program Analysis Platform." ACM Conference on Programming Language Design and Implementation (PLDI'11), June 2011.

6. Grants and Contracts

6.1. As Principal Investigator

1. Mayur Naik, Rajeev Alur, Insup Lee, Oleg Sokolsky, Boon Thau Loo. "ASPIRE: Automatically Subsetting Protocol Implementations Reliably and Efficiently". ONR #N00014-18-1-2021, $6,148,632, January 2018 (5

years).

2. Mayur Naik. "New Frontiers in Constraint-Based Program Analysis". NSF CCF #1526270, $450,000, September 2015 (3 years).
3. Mayur Naik, Molham Aref, Shan Shan Huang, Jens Palsberg, Tielei Wang. "PetaBlox: Large-Scale Software Analysis and Analytics Using Datalog". DARPA #FA8750-15-2-0009, $3,322,151, October 2014 (4 years).
4. Mayur Naik. "CAREER: Adaptive Large-Scale Program Analysis". NSF CCF #1253867, $484,402, January 2013 (5 years).
5. Mayur Naik. "Automated Scalable Testing of Mobile Apps Using Z3." Microsoft Software Engineering Innovation Foundation Award, $25,000, March 2012.
6. Mayur Naik. "Enhancing Mobile App Quality via Program Analysis." Google Faculty Research Award, $52,786, December 2011.

6.1.1. As Co-Principal Investigator

1. Rajeev Alur (PI), Insup Lee, Rahul Mangharam, Mayur Naik, Oleg Sokolsky, James Weimer. "REAFFIRM: Scenario-Based Design and Verification of Resilient Cyber-Physical Systems." DARPA #N66001-18-C-4007, $3,055,051, February 2018 (4 years).
2. Taesoo Kim (PI), Emmanouil Antonakakis, Mayur Naik, Wenke Lee. "Big Data and Security: Educating The Next-Generation Security Analysts." NSF DGE #1500084, $300,000, June 2015 (2 years).
3. Mostafa Ammar (PI), Irfan Issa, Mayur Naik, Ellen Zegura. "NeTS: Medium: Mobile Computing over Intermittently Connected Networks." NSF CNS #1161879, $695,000, August 2012 (3 years).
4. Hyesoon Kim (PI), Mayur Naik, Santosh Pande. "Smart Off-loading Algorithms and Profiling Techniques for Cloud Computing." Samsung, $185,000, May 2012 (1 year).
5. Alex Aiken (PI), Isil Dillig, Thomas Dillig, John Mitchell, Mayur Naik. "STAMP: STatic Analysis of Mobile Programs." DARPA #FA8750-12-2-0020, $3,929,766, January 2012 (4 years).

7. Software

1. *Petablox*, declarative program analysis framework for Big Code.
   Publicly available from https://github.com/petablox-project/.
2. *Nichrome*, solver for mixed hard and soft constraints.
   Publicly available from https://github.com/nichrome-project/.
3. *Dynodroid*, an automated input generation system for Android apps.
   Publicly available from https://github.com/dynodroid/dynodroid/.
4. *Acteve*, a dynamic symbolic execution engine for Android apps.
   Publicly available from https://bitbucket.org/pag-lab/acteve/.
5. *Chord*, a static and dynamic program analysis platform for Java programs.
   Publicly available from https://bitbucket.org/pag-lab/jchord/.
6. *CloneCloud*, a system for partitioning and migrating Android apps.
   (Not available publicly; proprietary of Intel Corporation.)
7. *CalFuzzer*, a randomized testing tool for concurrent Java programs.
   Publicly available from http://srl.cs.berkeley.edu/~ksen/calfuzzer/.
8. *CBI: Cooperative Bug Isolation*, a statistical-debugging framework for C programs.
   Publicly available from http://www.cs.wisc.edu/cbi/.

8. Teaching

8.1 Courses

Semester/Year	*Course Number*	*Course Title*	*Enrollment*	*Effectiveness (1-5)*	
				Instructor	*Course*
Spring 2015	CS 4240	Compilers and Interpreters	39	4.6	4.3
Fall 2014	CS 6340	Software Analysis and Testing	36	4.5	4.2
Spring 2014	CS 4400	Introduction to Database Systems	81	4.0	4.0
Fall 2013	CS 8803	Foundations of Programming Languages	11	4.3	4.3
Spring 2013	CS 4400	Introduction to Database Systems	42	4.2	4.2
Fall 2012	CS 8803	Foundations of Programming Languages	23	4.1	4.1
Fall 2011	CS 6340	Software Analysis and Testing	35	4.0	3.9

8.2 Curriculum Development

I developed a graduate-level course on software engineering titled "Software Analysis and Testing" that I teach regularly. The course covers the theory and practice of software analysis, which lies at the heart of many software development processes such as diagnosing bugs, testing, debugging, and more. It presents diverse techniques, each with their own strengths and limitations, for automating tasks such as testing, debugging, and finding bugs in complex real-world programs. These techniques include dataflow analysis, constraint-based analysis, type systems, model checking, symbolic execution, and more. The course teaches the principles underlying these techniques as well as imparts hands-on experience with using and implementing tools based on these techniques.

9. Students

1. **Halley Young**, 2017–present
2. **Richard Zhang**, 2017–present
3. **Xujie Si**, 2014–present
4. **Sulekha Kulkarni**, 2014–present
   Awards: Microsoft Research Graduate Women's Scholarship (2015)
5. **Xin Zhang**, 2011–2017
   Topic: Combining Logical and Probabilistic Reasoning in Program Analysis
   Awards: Georgia Tech College of Computing's Outstanding Graduate Research Award (2017), Facebook Fellowship (2015-2016), ACM SIGSOFT Distinguished Paper Award (FSE 2015), ACM SIGPLAN Distinguished Paper Award (PLDI 2014), Qualcomm Innovation Fellowship Finalist (2014)

10. Service

10.1 Conference Program Committee

- PC member, International Conference on Computer Aided Verification (CAV), 2018.
- PC member, Workshop on Forming an Ecosystem Around Software Transformation (FEAST), 2017.
- PC member, ACM Conference on Programming Language Design and Implementation (PLDI), 2017.
- PC member, International Static Analysis Symposium (SAS), 2016.
- ERC member, ACM Symposium on Principles of Programming Languages (POPL), 2016.
- Co-organizer, 4th International Workshop on the State Of The Art in Program Analysis (SOAP), 2015.

- PC member, Haifa Verification Conference (HVC), 2015.
- PC member, ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), 2015.
- PC member, ACM Symposium on Principles of Programming Languages (POPL), 2015.
- PC member, Haifa Verification Conference (HVC), 2014.
- PC member, India Software Engineering Conference (ISEC), 2014.
- PC member, ACM Workshop on Memory Systems Performance and Correctness (MSPC), 2013.
- PC member, ACM Conference on Programming Language Design and Implementation (PLDI), 2013.
- PC member, ACM Symposium on Principles and Practice of Parallel Programming (PPoPP), 2013.
- ERC member, ACM Symposium on Principles of Programming Languages (POPL), 2012.
- PC member, International Conference on Runtime Verification (RV), 2011.
- PC member, ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), 2011.
- PC member, New Ideas and Emerging Results (NIER) track, International Conference on Software Engineering (ICSE), 2010.
- ERC member, ACM Conference on Programming Language Design and Implementation (PLDI), 2010.

10.2 Journal Reviewing

- ACM Transactions on Software Engineering and Methodology, 2017
- ACM Transactions on Computer Systems, 2015
- ACM Transactions on Programming Languages and Systems, 2006, 2008–2010, 2015, 2016
- IEEE Transactions on Cloud Computing, 2015
- IEEE Transactions on Computers, 2009
- IEEE Transactions on Software Engineering, 2006