

# Aditya V. Thakur

February 2020

Address: University of California, Davis  
Department of Computer Science  
2063 Kemper Hall  
One Shields Avenue  
Davis, CA 95616-8562.  
Phone: +1 (530) 752-7004  
Email: avthakur@ucdavis.edu  
WWW: <http://thakur.cs.ucdavis.edu>

## Professional Preparation

University of Pune, India	Computer Engineering	B.E.	2004
Indian Institute of Science, Bangalore, India	Computer Science	M.Sc.(Engg.)	2008
University of Wisconsin, Madison	Computer Science	Ph.D.	2014

## Appointments

2017–present	Assistant Professor, University of California, Davis
2015–2017	Software Engineer, Google, Inc.
2014–2015	Assistant Research Engineer, Electrical Engineering and Computer Sciences Department, University of California, Berkeley, CA
2008–2014	Graduate Research Assistant, Computer Sciences Department, University of Wisconsin, Madison
2007–2008	Associate Researcher, Microsoft Research India
2005	Summer Research Intern, Microsoft Research, Redmond
2004–2007	Graduate Research Assistant, Supercomputer Education and Research Center, Indian Institute of Science, Bangalore, India

## Awards and Honours

Recipient of the Facebook Probability and Programming Research Award 2019.

Recipient of the Facebook Testing and Verification Research Award 2018.

Recipient of the 2018 Dean’s Collaborative Research (DECOR) Award from the College of Engineering, University of California, Davis.

Co-recipient of the Outstanding Graduate Student Research Award awarded by the Computer Sciences Department of the University of Wisconsin–Madison for 2013-2014.

Awarded 2013 Google PhD Fellowship in Programming Technology.

Selected as a Young Researcher to participate in the 1st Heidelberg Laureate Forum 2013.

## Publications

1. Kim, Sung Kook, Arnaud J. Venet, and **Aditya V. Thakur** (2020). Deterministic Parallel Fixpoint Computation. *Journal Proceedings of the ACM on Programming Languages PACMPL* 3(POPL).
2. DeFreez, Daniel, Cindy Rubio-Gonzalez, and **Aditya V. Thakur** (2019). Effective Error-Specification Inference via Domain-Knowledge Expansion. In: *Proceedings of the 27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE’19)*. ACM. DOI: 10.1145/3338906.3338960.
3. Ghosal, Dipak, Sambit Shukla, Alex Sim, **Aditya V. Thakur**, and Wu Kesheng (2019). A Reinforcement Learning Based Network Scheduler For Deadline-Driven Data Transfers. In: *IEEE Global Communications Conference, (GLOBECOM)*. IEEE.
4. Palakodety, Shriphani, Guha Jayachandran, and **Aditya V. Thakur** (2019). An SMT Approach To A Multiparty Economic Scheduling Problem. In: *17th International Workshop on Satisfiability Modulo Theories (SMT’19)*.

5. Sotoudeh, Matthew and **Aditya V. Thakur** (2019). Computing Linear Restrictions of Neural Networks. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*. <https://arxiv.org/pdf/1908.06214.pdf>.
6. Sotoudeh, Matthew and **Aditya V. Thakur** (2019). Correcting Deep Neural Networks with Small, Generalizing Patches. In: *NeurIPS 2019 Workshop on Safety and Robustness in Decision Making*.
7. DeFreez, Daniel, **Aditya V. Thakur**, and Cindy Rubio-González (2018). Path-Based Function Embedding and Its Application to Error-Handling Specification Mining. In: *26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2018)*. ACM, pp.423–433. doi: 10.1145/3236024.3236059.
8. Phothilimthana, Phitchaya Mangpo, **Aditya V. Thakur**, Rastislav Bodík, and D. Dhurjati (2016). Scaling up Superoptimization. In: *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. ACM, pp.297–310. doi: 10.1145/2872362.2872387.
9. Phothilimthana, Phitchaya Mangpo, **Aditya V. Thakur**, Rastislav Bodík, and Dinakar Dhurjati (2016). GreenThumb: superoptimizer construction framework. In: *Proceedings of the 25th International Conference on Compiler Construction (CC)*. ACM, pp.261–262. doi: 10.1145/2892208.2892233.
10. Reps, Thomas and **Aditya V. Thakur** (2016). Automating Abstract Interpretation. In: *17th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)*. Springer, pp.3–40. doi: 10.1007/978-3-662-49122-5\_1.
11. **Aditya V. Thakur**, Akash Lal, Junghee Lim, and Thomas W. Reps (2015). PostHat and All That: Automating Abstract Interpretation. *Electronic Notes in Theoretical Computer Science (ENTCS)* **311**. Fourth Workshop on Tools for Automatic Program Analysis (TAPAS 2013), 15–32.
12. **Aditya V. Thakur**, J. Breck, and Thomas Reps (2014). Satisfiability Modulo Abstraction for Separation Logic with Linked Lists. In: *2014 International Symposium on Model Checking of Software (SPIN)*. ACM, pp.58–67. doi: 10.1145/2632362.2632376.
13. Itzhaky, Shachar, Nikolaj Bjørner, Thomas W. Reps, Mooly Sagiv, and **Aditya V. Thakur** (2014). Property-Directed Shape Analysis. In: *26th International Conference on Computer Aided Verification (CAV)*. Springer, pp.35–51. doi: 10.1007/978-3-319-08867-9\_3.
14. **Aditya V. Thakur**, Matt Elder, and Thomas Reps (2012). Bilateral Algorithms for Symbolic Abstraction. In: *19th International Symposium on Static Analysis (SAS)*. Springer, pp.111–128. doi: 10.1007/978-3-642-33125-1\_10.
15. **Aditya V. Thakur** and Thomas Reps (2012). A Generalization of Stålmarck’s Method. In: *19th International Symposium on Static Analysis (SAS)*. Springer, pp.334–351. doi: 10.1007/978-3-642-33125-1\_23.
16. **Aditya V. Thakur** and Thomas Reps (2012). A Method for Symbolic Computation of Precise Abstract Operations. In: *24th International Conference on Computer Aided Verification (CAV)*. Springer, pp.174–192. doi: 10.1007/978-3-642-31424-7\_17.
17. Driscoll, Evan, **Aditya V. Thakur**, and Thomas Reps (2012). OpenNWA: A Nested-Word Automaton Library. In: *24th International Conference on Computer Aided Verification (CAV)*. Springer, pp.665–671. doi: 10.1007/978-3-642-31424-7\_47.
18. **Aditya V. Thakur**, Junghee Lim, Akash Lal, Amanda Burton, Evan Driscoll, Matt Elder, Tycho Andersen, and Thomas Reps (2010). Directed Proof Generation for Machine Code. In: *22nd International Conference on Computer Aided Verification (CAV)*. Springer, pp.288–305. doi: 10.1007/978-3-642-14295-6\_27.
19. Beckman, Nels, Aditya Nori, Sriram Rajamani, Robert Simmons, Sai Deep Tetali, and **Aditya V. Thakur** (2010). Proofs from Tests. *IEEE Transactions on Software Engineering (TSE)* **36**(4), 495–508.
20. Jin, Guoliang, **Aditya V. Thakur**, Ben Liblit, and Shan Lu (2010). Instrumentation and Sampling Strategies for Cooperative Concurrency Bug Isolation. In: *Proceedings of the 25th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*. ACM, pp.241–255. doi: 10.1145/1869459.1869481.
21. Reps, Thomas, Junghee Lim, **Aditya V. Thakur**, Gogul Balakrishnan, and Akash Lal (2010). There’s Plenty of Room at the Bottom: Analyzing and Verifying Machine Code. In: *22nd International Conference on Computer Aided Verification (CAV)*. Invited paper. Springer, pp.41–56. doi: 10.1007/978-3-642-14295-6\_6.
22. **Aditya V. Thakur**, Rathijit Sen, Ben Liblit, and Shan Lu (2009). Cooperative Crug Isolation. In: *Seventh International Workshop on Dynamic Analysis (WODA)*. ACM, pp.35–41. doi: 10.1145/2134243.2134252.
23. Nori, Aditya, Sriram Rajamani, Sai Deep Tetali, and **Aditya V. Thakur** (2009). The Yogi Project: Software Property Checking via Static Analysis and Testing. In: *15th International Conference on Tools and Algorithms*

for the Construction and Analysis of Systems (TACAS). Springer, pp.178–181. doi: 10.1007/978-3-642-00768-2\_17.

24. **Aditya V. Thakur** and R. Govindarajan (2008). Comprehensive Path-sensitive Data-flow Analysis. In: *Sixth International Symposium on Code Generation and Optimization (CGO)*. ACM, pp.55–63. doi: 10.1145/1356058.1356066.

### Ph.D. Dissertation

1. **Aditya V. Thakur** (2014). “Symbolic Abstraction: Algorithms and Applications”. <http://digital.library.wisc.edu/1793/69652>. PhD thesis. Computer Sciences Department, University of Wisconsin, Madison.

### Master’s Thesis

1. **Aditya V. Thakur** (2008). “Comprehensive Path-sensitive Data-flow Analysis”. M.Sc.(Engg.) thesis. Indian Institute of Science, Bangalore.

## Presentations

1. *Code Embeddings for Bug Finding* (2019). Probability and Programming Workshop, Big Code Summit, PLEMM Summit. Seattle, Washington, USA.
2. *Abstract Interpretation and Decision Procedures: Better Together!* (2014). University of California, Berkeley, USA.
3. *More Algorithms for Symbolic Abstraction* (2014). Dagstuhl Seminar 14351 Decision Procedures and Abstract Interpretation. Schloss Dagstuhl, Leibniz Center for Informatics, Germany.
4. *Through the Lens of Abstraction* (2014). Fourteenth Annual High Confidence Software and Systems Conference, Annapolis, MD, USA.
5. *Through the Lens of Abstraction* (2014). Dagstuhl Seminar 14241 Challenges in Analysing Executables: Scalability, Self-Modifying Code and Synergy. Schloss Dagstuhl, Leibniz Center for Informatics, Germany.
6. *PostHat and All That: Attaining Most-Precise Inductive Invariants* (2013). Fourth Workshop on Tools for Automatic Program Analysis (TAPAS). Seattle, WA, USA.
7. *A Deductive Algorithm for Symbolic Abstraction with Applications to SMT* (2012). Computer Science Dept., Stanford University, CA, USA.
8. *A Deductive Algorithm for Symbolic Abstraction with Applications to SMT* (2012). Max Planck Institute for Software Systems, Kaiserslautern, Germany.
9. *A Generalization of Stålmarck’s Method* (2012). 19th International Symposium on Static Analysis (SAS). Deauville, France.
10. *A Method for Symbolic Computation of Abstract Operations* (2012). International Conference on Computer Aided Verification (CAV). Berkeley, CA, USA.
11. *A Method for Symbolic Computation of Abstract Operations* (2012). Dagstuhl Seminar 12051 Analysis of Executables: Benefits and Challenges. Schloss Dagstuhl, Leibniz Center for Informatics, Germany.
12. *Bilateral Algorithms for Symbolic Abstraction* (2012). 19th International Symposium on Static Analysis (SAS). Deauville, France.
13. *Cooperative Crug Isolation* (2009). Seventh International Workshop on Dynamic Analysis (WODA). Chicago, IL, USA.
14. *Cooperative Crug Isolation* (2009). Microsoft Research India, Bangalore, India.
15. *Cooperative Crug Isolation* (2009). Indian Institute of Science, Bangalore, India.
16. *Comprehensive Path-sensitive Data-flow Analysis* (2008). Sixth International Symposium on Code Generation and Optimization (CGO). Boston, MA, USA.

## Professional Activities

Program Committee member

International Conference on Computer Aided Verification (CAV), 2020.

Journal First Presentations Track, International Conference on Automated Software Engineering (ASE), 2019.

26th Static Analysis Symposium (SAS), 2019.

11th Conference on Verified Software: Theories, Tools, and Experiments (VSTTE), 2019.

International Conference on Automated Software Engineering (ASE), 2018.

International Symposium on Automated Technology for Verification and Analysis (ATVA), 2018.

External Review Committee, International Conference on Programming Language Design and Implementation (PLDI), 2016.

Reviewer, National Science Foundation (NSF) panel, 2019.

Reviewer, National Science Foundation (NSF) panel, 2018.

Coordinator of Dagstuhl seminar 14351 *Decision Procedures and Abstract Interpretation* at Schloss Dagstuhl-Leibniz Center for Informatics (Wadern, Germany) held in August 2014.

Member, Association for Computing Machinery, 2008–Present.

Reviewer for the following research publications:

International Conference on Principles of Programming Languages, 2019

ACM Transactions on Software Engineering and Methodology (TOSEM), 2016

Static Analysis Symposium (SAS), 2010, 2011

Certified Programs and Proofs (CPP), 2011

European Symposium on Programming (ESOP), 2012

Higher-Order and Symbolic Computation (HOSC), 2011

Verification, Model Checking, and Abstract Interpretation (VMCAI), 2011, 2013

Computer and Communications Security (CCS), 2011

Formal Methods in Computer-Aided Design (FMCAD), 2012

Formal Methods in Systems Design (FMSD), 2012

Computer Aided Verification (CAV), 2013

Information and Computation, 2013

International Journal on Software Tools for Technology Transfer (STTT), 2014

## Teaching

Instructor, University of California, Davis

- Programming Languages (Graduate-level course)
- Topics in Automated Reasoning and Program Analysis (Graduate-level course)
- Introduction of Programming Languages (Undergraduate-level course)

Teaching assistant, University of Wisconsin–Madison

- Principles Of Programming Languages (Graduate-level course)
- Introduction to Programming Languages and Compilers (Undergraduate-level course)