Aditya V. Thakur

August 2023

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

2022–present	Associate Professor, University of California, Davis
2017–2022	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 DOE Early Career Award 2021.

Recipient of NSF CAREER Award 2021.

Recipient of the Facebook Probability and Programming Research Award 2020.

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.

Peer-Reviewed Publications

1. Matthew Sotoudeh, Zhe Tao and **Thakur, Aditya V.** (2023). SyReNN: A Tool for Analyzing Deep Neural Networks. *International Journal on Software Tools for Technology Transfer (STTT)*.
2. Tao, Zhe, Stephanie Nawas, Jacqueline Mitchell, and **Thakur, Aditya V.** (2023). Architecture-Preserving Provable Repair of Deep Neural Networks. *Proc. ACM Program. Lang.* **7**(PLDI).
3. Sotoudeh, Matthew and **Aditya V. Thakur** (2021). Provable Repair of Deep Neural Networks. In: *42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI)*. ACM.

4. Sotoudeh, Matthew and **Aditya V. Thakur** (2021). SyReNN: A Tool for Analyzing Deep Neural Networks. In: *27th International Conference on Tools and Algorithms for the Construction and Analysis of Systems TACAS*. Springer. DOI: 10.1007/978-3-030-72013-1_15.

5. Tao, Zhe, Aseem Rastogi, Naman Gupta, Kapil Vaswani, and **Aditya V. Thakur** (2021). DICE*: A Formally Verified Implementation of DICE Measured Boot. In: *30th USENIX Security Symposium (USENIX Security '21 Fall)*. https://www.usenix.org/conference/usenixsecurity21/presentation/tao.

6. Kim, Sung Kook, Arnaud J. Venet, and **Aditya V. Thakur** (2020). Deterministic Parallel Fixpoint Computation. *Journal Proceedings of the ACM on Programming Languages (PACMPL)* 4(POPL), 14:1–14:33.

7. Kim, Sung Kook, Arnaud J. Venet, and **Aditya V. Thakur** (2020). Memory-efficient Fixpoint Computation. In: *27th Static Analysis Symposium (SAS)*. **Awarded the Radhia Cousot Young Researcher Best Paper Award.** DOI: 10.1007/978-3-030-65474-0_3.

8. R. Ghosal, Gaurav, Dipak Ghosal, Alex Sim, **Aditya V. Thakur**, and Kesheng Wu (2020). A Deep Deterministic Policy Gradient Based Network Scheduler For Deadline-Driven Data Transfer. In: *2020 IFIP Networking Conference*.

9. Sotoudeh, Matthew and **Aditya V. Thakur** (2020). Abstract Neural Networks. In: *27th Static Analysis Symposium (SAS)*. DOI: 10.1007/978-3-030-65474-0_4.

10. Sotoudeh, Matthew and **Aditya V. Thakur** (2020). Analogy-Making as a Core Primitive in the Software Engineering Toolbox. In: *2020 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Onward!* ACM. DOI: 10.1145/3426428.3426918.

11. 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.

12. 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. DOI: 10.1109/GLOBECOM38437.2019.9013255.

13. 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)*.

14. 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)*. http://papers.nips.cc/paper/9562-computing-linear-restrictions-of-neural-networks.pdf.

15. 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*.

16. 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.

17. DeFreez, Daniel, **Aditya V. Thakur**, and Cindy Rubio-González (2018). Path-Based Function Embeddings. In: *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings, ICSE 2018*. ACM, pp.430–431. DOI: 10.1145/3183440.3195042.

18. 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.

19. 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.

20. 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.

21. **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.

22. **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.

23. 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.

24. **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.

25. **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.

26. **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.

27. 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.

28. **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.

29. 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.

30. 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.

31. 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.

32. **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.

33. 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.

34. **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. *Towards Trustworthy Scientific Machine Learning* (2023). DOE/NSF Workshop on Correctness in Scientific Computing (CSC). Florida, U.S.A.

2. *Provable Repair of Deep Neural Networks* (2021). 4th Workshop on Formal Methods for ML-Enabled Autonomous Systems Affiliated with CAV 2021. (virtual).

3. *Provable Repair of Deep Neural Networks* (2021). Tenth Summer School on Formal Techniques. (virtual).

4. *Efficient Fixpoint Computation for Abstract Interpretation* (2020). Workshop on Research Highlights in Programming Languages. Goa, India (virtual).

5. *Understanding and Repairing Deep Neural Networks* (2020). Waterloo ML+Logic Colloquium. University of Waterloo, Canada (virtual).

6. *Code Embeddings for Bug Finding* (2019). Probability and Programming Workshop, Big Code Summit, PLEMM Summit. Seattle, Washington, USA.

7. *Abstract Interpretation and Decision Procedures: Better Together!* (2014). University of California, Berkeley, USA.

8. *More Algorithms for Symbolic Abstraction* (2014). Dagstuhl Seminar 14351 Decision Procedures and Abstract Interpretation. Schloss Dagstuhl, Leibniz Center for Informatics, Germany.

9. *Through the Lens of Abstraction* (2014). Fourteenth Annual High Confidence Software and Systems Conference, Annapolis, MD, USA.

10. *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.

11. *PostHat and All That: Attaining Most-Precise Inductive Invariants* (2013). Fourth Workshop on Tools for Automatic Program Analysis (TAPAS). Seattle, WA, USA.

12. *A Deductive Algorithm for Symbolic Abstraction with Applications to SMT* (2012). Computer Science Dept., Stanford University, CA, USA.

13. *A Deductive Algorithm for Symbolic Abstraction with Applications to SMT* (2012). Max Planck Institute for Software Systems, Kaiserslautern, Germany.

14. *A Generalization of Stålmarck's Method* (2012). 19th International Symposium on Static Analysis (SAS). Deauville, France.

15. *A Method for Symbolic Computation of Abstract Operations* (2012). International Conference on Computer Aided Verification (CAV). Berkeley, CA, USA.

16. *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.

17. *Bilateral Algorithms for Symbolic Abstraction* (2012). 19th International Symposium on Static Analysis (SAS). Deauville, France.

18. *Cooperative Crug Isolation* (2009). Seventh International Workshop on Dynamic Analysis (WODA). Chicago, IL, USA.

19. *Cooperative Crug Isolation* (2009). Microsoft Research India, Bangalore, India.

20. *Cooperative Crug Isolation* (2009). Indian Institute of Science, Bangalore, India.

21. *Comprehensive Path-sensitive Data-flow Analysis* (2008). Sixth International Symposium on Code Generation and Optimization (CGO). Boston, MA, USA.

Grants

co-PI, *SHF:Medium: Studying and Exploiting the Bimodality of Software*, NSF, 2021–2024.

PI, *AutoNeurify: Automatic Infusion of Learning in HPC Applications*, DOE, 2021–2026.

PI, *CAREER: Provable Repair of Deep Neural Networks*, NSF, 2021–2026.

PI, *Provable polytope patching of deep neural networks*, Facebook, 2020–2021.

PI, *Code Embeddings for Bug Finding*, Facebook, 2019-2020.

PI, *Efficient Concurrent Abstract Interpretation*, Facebook, 2018–2019.

co-PI, *Effective Desulfurization Through Microaeration During Anaerobic Digestion*, College of Engineering, University of California, Davis, 2018–2019.

Professional Activities

Program Committee member

International Conference on Computer-Aided Verification (CAV), 2023

International Conference on Programming Language Design and Implementation (PLDI), 2022

International Conference on Principles of Programming Languages (POPL), 2022

International Symposium on Automated Technology for Verification and Analysis (ATVA), 2021

IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), 2020.

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.

ACM SIGPLAN International Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), 2020

Reviewer, Department of Energy (DOE) panel, 2021.

Reviewer, National Science Foundation (NSF) panel, 2021 (three separate panels).

Reviewer, National Science Foundation (NSF) panel, 2020 (two panels).

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

Reviewer for the following research publications:

International Conference on Principles of Programming Languages (POPL), 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

Reviewer for the book *Shape Analysis* published as part of book series on Foundations and Trends in Programming Language, 2020.

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.

Graduate Student Advising

Nishant Acharya (Ph.D.), 2023–present
Sameerah Helal (Ph.D.), 2022–present
Stephanie Nawas (Ph.D.), 2021–present
Zhe Tao (Ph.D.), 2019–present
Sung Kook Kim (Ph.D.), 2018–2020

- Awarded Radhia Cousot Young Researcher Best Paper Award at SAS 2020.
- Runner Up for the Graduate Group for Computer Science (GGCS) 2020 Best Graduate Researcher Award, UC Davis.

Daniel DeFreez (Ph.D., 2019),

- First position: Assistant Professor, Southern Oregon University.
- Co-advised with Prof. Cindy Rubio González.
- Awarded first place in ESEC/FSE 2018 Student Research Competition (SRC).

Mohamed Alkaoud (M.S., 2018)
Jonathan Castello (M.S., 2018)
Dana Iltis (M.S., 2018)

Undergraduate Student Advising

Jacqueline Mitchell 2021–present
Daniel Phan, 2020-2021
Matthew Sotoudeh, 2019–2021

- Received an Honorable Mention for 2021 Chancellors Award for Excellence in Undergraduate Research/Dean Keith Simonton Prize, UC Davis.
- Received an NSF GRFP 2021.
- Received Honorable Mention in the 2021 CRA Outstanding Undergraduate Researcher Award.
- Awarded second place in POPL 2020 Student Research Competition.

Eric Li, 2019-2021
Jack Abukhovski, 2019-2020
Ashley Li, 2020
James Sun, 2017-2019
Scott Reichelt, 2017-2019
Michael Field, 2019
Max Nedorezov, 2017-2018
Zaid Al Rakabi, 2018
Xinyuan Sun, 2017-2018

Teaching

Instructor, University of California, Davis

- Programming Languages (Graduate course), Winter 2023, Fall 2020, Fall 2019, Fall 2018
- Topics in Automated Reasoning and Program Analysis (Graduate course), Winter 2020, Fall 2017
- Introduction of Programming Languages (Undergraduate course), Fall 2022, Winter 2021, Winter 2019, Spring 2018

Teaching assistant, University of Wisconsin–Madison

- Principles Of Programming Languages (Graduate course), Spring 2010
- Introduction to Programming Languages and Compilers (Undergraduate course), Fall 2008