Divyesh Unadkat

+91 928 450 2604
divyeshunadkat.github.io
divyeshunadkat

Education

Ph.D. — **Computer Science & Engineering**, *Indian Institute of Technology Bombay*, Mumbai. **2023**
CPI: 9.48/10

B.E. — **Computer Engineering**, *Dharmsinh Desai University*, Nadiad. **2006 – 2010**
Aggregate: 80.12 %

Ph.D. Thesis

Title: *Techniques for Precise and Scalable Verification of Array Programs*

Supervisors: Prof. Supratik Chakraborty , Prof. Ashutosh Kumar Gupta

Institution: Indian Institute of Technology Bombay, Mumbai

Area: Formal Methods and Software Verification

Experience

Senior Staff R&D Engineer, *Synopsys*, Hyderabad.	**Feb'24–Present**
Staff R&D Engineer, *Synopsys*, Hyderabad.	**Aug'23–Jan'24**
Scientist/Senior Software Engineer, *TCS Research*, Pune.	**Jun'21–Jul'23**
Researcher/Software Engineer, *TCS Research*, Pune.	**Jun'10–May'21**
Research Intern, *TCS Research*, Pune.	**Dec'09–Apr'10**

Technical Skills

Programming: C++, C, Java, Python, LaTeX

Compilers: LLVM, Clang, GNU Tool Chain (GCC, GDB, Make)

Research Tools: Z3, CVC5, CBMC, Daikon, CPAChecker, InvGen

Development Tools: Visual Studio Code, Emacs, Vim, Eclipse

Version Control: Git, Perforce, CVS

Telecommunication Tools: Google Hangouts, Microsoft Teams, Zoom, Slack

OS: Linuix, Windows.

Tool Dev

Diffy	Generalized Inductive Reasoning for Arrays. Published in CAV 2021 [3]. *repository*
Vajra	Full-Program Induction. Published in TACAS 2020 [4, 5], STTT 2022 [2]. *repository*
Tiler	Verifying Array Programs by Tiling. Published in SAS 2017 [6]. repository
DIV	Dynamic Inference Verifier. Internal Tool, TCS Research. Published in HVC 2013 [8]
ScaleM	Scaling Model Checking with Abstractions Inferred using Dynamic Analysis. Internal Tool, TCS Research. Published in ICST 2013 [7]
AutoGen	Automatic Test-case Generation using Model Checking. Internal Tool, TCS Research

Publications

[1] Divyesh Unadkat. Techniques for Precise and Scalable Verification of Array Programs. *Doctoral Dissertation, IIT Bombay*, December 2022.

[2] Supartik Chakraborty, Ashutosh Gupta, and Divyesh Unadkat. Full-Program Induction: Verifying Array Programs sans Loop Invariants. In *International Journal on Software Tools for Technology Transfer (STTT)*, pages 843–888, September 2022.

[3] Supartik Chakraborty, Ashutosh Gupta, and Divyesh Unadkat. Diffy: Inductive Reasoning of Array Programs using Difference Invariants. In *Proc. of the 33rd International Conference on Computer-Aided Verification (CAV)*, pages 911–935, 2021.

[4] Supartik Chakraborty, Ashutosh Gupta, and Divyesh Unadkat. Verifying Array Manipulating Programs with Full-Program Induction. In *Proc. of the 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 22–39, 2020.

[5] Mohammad Afzal et. al. VeriAbs : Verification by Abstraction and Test Generation (Competition Contribution). In *Proc. of the 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 383–387, 2020.

[6] Supratik Chakraborty, Ashutosh Gupta, and Divyesh Unadkat. Verifying Array Manipulating Programs by Tiling. In *Proc. of the 24th International Static Analysis Symposium (SAS)*, pages 428–449, 2017.

[7] Anand Yeolekar et. al. Scaling Model Checking for Test Generation using Dynamic Inference. In *Proc. of the 6th International Conference on Software Testing, Verification and Validation (ICST)*, pages 184–191, 2013.

[8] Anand Yeolekar and Divyesh Unadkat. Assertion Checking using Dynamic Inference. In *Proc. of the 9th Haifa Verification Conference (HVC)*, pages 199–213, 2013.

Awards

Individual Award: Spot Award

Institution: Synopsys, Hyderabad

Description: Performance par excellence. FY 2024.

Individual Award: Conference Travel Grant

Institution: ACM India

Description: Selected among only 50 national and international participants for the inaugural edition of ACM Pingla Interactions in Computing 2024 held at Infosys Mysore.

Team Award (Recurring): Best Verification Tool

Institution: International Software Verification Competition (SV-COMP)

Description: VeriAbs [5] won SV-COMP in 2022, 2021, and 2020. My contribution included the development of array verification tools Diffy [3], Vajra [4, 2] and Tiler [6] which are integrated in VeriAbs. Refer to [5] and [1] for more details.

Individual Award: Special Mention on CSE IITB Webpage

Institution: Indian Institute of Technology Bombay, Mumbai

Description: SV-Comp win acknowledged on CSE IITB news page. Posted on 22-12-2019.

Individual Award: Most Admired Sprint Thesis Talk

Institution: Indian Institute of Technology Bombay, Mumbai

Description: Runner-up, Senior Researcher Sprint Talks, RISC 2017, IIT Bombay.

Individual Award: Best Speaker in Sprint Thesis Talk

Institution: Indian Institute of Technology Bombay, Mumbai

Description: Winner, Early Researcher Sprint Talks, RISC 2016, IIT Bombay.

Individual Award: Eklavya Gold Medal

Institution: Dharmsinh Desai University, Nadiad

Description: Highest aggregate in first four semesters of computer engineering, 2008.

Projects

Generalized Full-Program Induction

We devised an inductive reasoning technique that further generalizes the *full-program induction* technique. Significantly, we simplify the inductive step of the analysis. The method can prove a sub-class of quantified as well as quantifier-free assertions in array programs with sequentially composed as well as nested loops. Like full-program induction, this technique also inducts over the entire program via the program parameter N and infers *difference invariants* between two slightly different versions of a program during the inductive step. We have implemented it in the tool *Diffy* and demonstrated its effectiveness vis-a-vis award winning verification tools on a set of array benchmarks from SV-COMP. Refer publications [1, 3] for details.

Timeline: 2020 – 2021

Language/Tool: C++/LLVM, SMTLIB2/Z3

Team size: One

Full-Program Induction

Formally verifying properties of programs that manipulate arrays of parametric size in loops is computationally challenging. In this project, we designed the novel *Full-Program Induction* technique to overcome this challenge. The technique inducts over the entire program via the program parameter N and computes the difference of programs and properties with different parameter values during the inductive step. We have implemented it in the tool *Vajra*. We compare its performance vis-a-vis state-of-the-art verification tools on a set of array manipulating benchmarks. Refer publications [1, 2, 4, 5] for details.

Timeline: 2018 – 2020

Language/Tool: C++/LLVM, SMTLIB2/Z3

Team size: Two

Verification by Tiling

We developed a novel property-driven verification method that can infer array access patterns in loops, and use this information to compositionally prove universally quantified assertions about arrays. We have implemented this technique in a tool called *Tiler*, written in C++, which outperforms several state-of-the-art tools on a suite of interesting benchmarks. Refer publication [1, 6] for details.

Timeline: 2016 – 2017

Language/Tool: C++/Clang, Java/Daikon, SMTLIB2/Z3

Team size: One

Dynamic Inference and Verification

Model checkers often run into scalability issues when verifying programs with a large state space. To address this challenge, several abstraction-based techniques have been proposed in the literature. We have developed an innovative CEGAR technique that generates refinement predicates using dynamic analysis. Our verification approach, implemented in *DIV*, using the Java programming language, is sound and compositional, even though the underlying dynamic analysis may return unsound predicates. We have demonstrated its effectiveness on challenging benchmarks. Refer publication [8] for details.

Timeline: 2013 – 2014

Language/Tool: Java/Daikon, C++/Kvasir, C++/CBMC

Inhouse Tools: C++/(AutoGen, Misra-CFE), Java/(PRISM, Slicer, Unparser)

Team size: Two

Scaling Model Checking

Automatic testcase generation tools seldom scale to large systems. We proposed a new approach that uses source-level abstractions generated using dynamic analysis to abstract parts of code to scale the testcase generation effort based on model checking. Our tool *ScaleM*, written in Java, summarizes complex code fragments to enable the generation of application-level tests in large-size C code using inferred program properties. Refer publication [7] for details.

Timeline: 2012 – 2013

Language/Tool: Java/Daikon, C++/Kvasir, C++/CBMC

Inhouse Tools: C++/(AutoGen, Misra-CFE), Java/(PRISM, Slicer, Unparser)

Team Size: Three

Automatic Testcase Generation

Rigorous verification of safety critical systems is inevitable. Manual testing of such systems is inefficient as well as inadequate. We have developed *AutoGen*, written in C++, a fully automatic, structural testcase generation tool for C programs that uses a model checker at the back-end. The tool is capable of generating testcases satisfying various coverage criteria including the modified condition/decision coverage (MC/DC) mandated by ISO 26262 & DO178B standards.

Timeline: 2010 – 2011

Language/Tool: C++/CBMC

Inhouse Tools: C++/Misra-CFE, Java/(PRISM, Slicer, Unparser)

Team size: Five

Statechart Translation

Harel statecharts are widely used to model real time reactive systems using the Statemate tool. During my internship, we developed a tool for translating various statecharts into Symbolic Analysis Laboratory (SAL) specification. Statecharts are first converted into an Intermediate Representation (IR). We then unparse the IR to SAL specification. The motive was to enable verification of statecharts for erroneous conditions like Non Determinism, Race Conditions and State Reachability using the relatively more scalable model checker available in the SAL framework, sal-bmc.

Timeline: 2009 – 2010

Tools & Languages: SAL, Statemate, lex, yacc, bison

Inhouse Tool: ACH-SCH Parser

Team size: Two

Conference Presentations

Diffy: Verifying Array Programs using Difference Invariants: 33rd International Conference on Computer Aided Verification (CAV), Los Angeles, USA (*Online*), July 2021

Verifying Array Manipulating Programs with Full-Program Induction: 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Luxembourg (*Online*), March 2021

Verifying Array Manipulating Programs by Tiling: 24th International Static Analysis Symposium, SAS, New York, USA, August 2017

Assertion Checking using Dynamic Inference: 9th Haifa Verification Conference, Haifa, Israel, November 2013

Invited Talks

Dance of the Dragons: Induction, Difference Computation and SMT Solving: Formal Methods Update Meeting, IIT Delhi, July 2022

Difference Invariants for Inductive Verification: 6th Indian SAT+SMT School (*Online*), December 2021

Exploiting Induction and Difference Computation to Verify Array Programs: Formal Methods Update Meeting (*Online*), July 2021

The Full-Program Induction Technique: 5th Indian SAT+SMT School, IIT Hyderabad (*Online*), December 2020

Verifying Array Manipulating Programs with Full-Program Induction: Software Engineering Research India (SERI), IIIT Hyderabad (*Online*), July 2020

Lightening Talk: Verifying Array Manipulating Programs by Tiling: 2nd Indian SAT+SMT School, Infosys Campus, Mysuru, December 2017

Competition Talks

Verifying Array Manipulating Programs by Full-Program Induction: Research and Innovation Symposium in Computing, RISC 2019, IIT Bombay

Verifying Array Manipulating Programs by Tiling: Sprint Thesis Talk, Research and Innovation Symposium in Computing, RISC 2017, IIT Bombay

Towards Precise Software Verification: Sprint Thesis Talk, Research and Innovation Symposium in Computing, RISC 2016, IIT Bombay

Poster Presentations

Full-Program Induction: A paradigm shift in the Verification of Array Programs: ACM Pingla Interactions in Computing, Infosys, Mysore, February 2024

Verifying Array Programs with Full-Program Induction: 4th Indian SAT+SMT School, IIT Bombay, December 2019

Executive Summary on Tiling to Verify Array Programs : TCS Anvetion Workshop, IIT Madras Research Park, Chennai, 2018

Verifying Array Manipulating Programs by Tiling: Research and Innovation Symposium in Computing, RISC, IIT Bombay, Mumbai, March 2017

Interests

Sports: Table Tennis, Volleyball, Football

Recreation: Yoga, Novels, Music, Movies

Links

Webpage: https://divyeshunadkat.github.io/

LinkedIn: https://www.linkedin.com/in/divyeshunadkat/

GitHub: https://github.com/divyeshunadkat/

dblp: https://dblp.uni-trier.de/pers/hd/u/Unadkat:Divyesh

Scholar: https://scholar.google.co.in/citations?user=8d48NqMAAAAJ

Contact

Mobile: +91 928 450 2604

E-Mail: divyeshunadkat001@gmail.com

References

Available upon request.