(+91) 9284502604

Divyesh Unadkat

Education

Ph.D. in Computer Science and Engineering (pursuing), **2015–Present**
Indian Institute of Technology Bombay, Mumbai.
CPI: 9.48/10

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

Experience

Researcher, *Tata Research Development and Design Centre*, Pune. **Jun'10–Present**

Intern, *Tata Research Development and Design Centre*, Pune. **Dec'09–Apr'10**

Technical Skills

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

Development Tools: Emacs, Vim, Eclipse, Cygwin, Git, CVS

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

Compilers: LLVM, Clang, GNU tool chain (GCC, GDB, Make)

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

Publications

[1] Mohammad Afzal, Supratik Chakraborty, Avriti Chauhan, Bharti Chimdyalwar, Priyanka Darke, Ashutosh Gupta, Shrawan Kumar, Charles Babu M, Divyesh Unadkat, and R. Venkatesh. Veriabs : Verification by abstraction and test generation (competition contribution). In *Proceedings of the 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 383–387, 2020.

[2] Supartik Chakraborty, Ashutosh Gupta, and Divyesh Unadkat. Verifying array manipulating programs with full-program induction. In *Proceedings of the 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 22–39, 2020.

[3] Supratik Chakraborty, Ashutosh Gupta, and Divyesh Unadkat. Verifying array manipulating programs by tiling. In *Proceedings of the 24th International Static Analysis Symposium (SAS)*, pages 428–449, 2017.

[4] Anand Yeolekar and Divyesh Unadkat. Assertion checking using dynamic inference. In *Proceedings of the 9th Haifa Verification Conference (HVC)*, pages 199–213, 2013.

[5] Anand Yeolekar, Divyesh Unadkat, Vivek Agarwal, Shrawan Kumar, and R Venkatesh. Scaling model checking for test generation using dynamic inference. In *Proceedings of the 6th International Conference on Software Testing, Verification and Validation (ICST)*, pages 184–191, 2013.

Presentations & Posters

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

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

Executive Poster: Verifying Array Programs by Tiling: TCS Anvetion Workshop, **IITM Research Park, Chennai**, 2018

Presentations & Posters

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

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

Competition Talk: Verifying Array Manipulating Programs by Tiling: Research and Innovation Symposium in Computing, RISC, **IIT Bombay**, April 2017.

Competition Talk: Towards Precise Software Verification: Research and Innovation Symposium in Computing, RISC, **IIT Bombay**, April 2016.

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

Tools/Projects

VAJRA

Formally verifying properties of programs that manipulate arrays of parametric size in loops is computationally challenging. In this project, we have developed the *Full-Program Induction* technique for proving (a sub-class of) quantified as well as quantifier-free assertions in such programs. In this technique, we induct over the entire program via the program parameter N. We have demonstrated the effectiveness of our tool VAJRA vis-a-vis several verification tools on a set of array manipulating benchmarks. Publications [2, 1]

TILER

In the TILER project, we have 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 our method in a tool called TILER which outperforms several state-of-the-art tools on a suite of interesting benchmarks. Publication [3]

DIV

Model checkers run into scalability issues when verifying programs with large state spaces. To tackle this situation, several abstraction-based techniques have been proposed. We have developed an innovative CEGAR style, based on the predicates generated using dynamic analysis. Our verification approach is sound and compositional, even though dynamic analysis may return unsound predicates. We have demonstrated its effectiveness on challenging benchmarks. Publication [4]

ScaleM

Automatic test generation tools hardly scale to large systems code. We proposed a new approach that uses source level abstractions generated using dynamic analysis to abstract parts of code. Our tool ScaleM summarizes complex code fragments to enable application level test generation on large size C code using inferred program properties. Publication [5]

AutoGen

Rigorous verification of safety critical systems is inevitable. Manual testing of such systems is inefficient as well as inadequate. We have developed AutoGen, a model checking based, fully automatic, structural test case generation tool for C programs. The tool is capable of generating test cases satisfying various coverage criteria including modified condition/decision coverage (MC/DC) mandated by standards - ISO 26262 & DO178B.

Statechart Analyzer

Harel state charts are widely used to model real time reactive systems using Statemate tool. During my internship, we developed a tool for translating various statechart formalisms to Symbolic Analysis Laboratory (SAL) specification. The motive was to enable verification of state charts for erroneous conditions like Non Determinism, Race Conditions and State Reachability using sal-bmc, the model checker available in the SAL framework.

Awards

Award: Best Verification Tool

Institution: International Software Verification Competition (SV-COMP)

Description: VeriAbs stood first in the ReachSafety category at SV-COMP 2020. As a part of its team, I designed and developed an induction-based verification techniques for Arrays sub-category, and implemented them in tools Vajra and Tiler as well as integrated them with VeriAbs [1].

Award: Most Admired Sprint Thesis Talk

Institution: Indian Institute of Technology Bombay, Mumbai

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

Award: Best Speaker in Sprint Thesis Talk

Institution: Indian Institute of Technology Bombay, Mumbai

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

Award: Eklavya Gold Medal

Institution: Dharmsinh Desai University, Nadiad

Description: Highest aggregate marks in first four semesters, CE 2008, DDU Nadiad.

Contact

Mobile: (+91) 9284502604

E-Mail: divyesh@cse.iitb.ac.in

Links

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

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

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

References

Available upon request.