

Ashish Mishra

No-480
New Men's Hostel, IISc, Bangalore
☎ +91 9611194714
✉ ashishmishra@csa.iisc.ernet.in
🌐 www.clweb.csa.iisc.ernet.in/ashishmishra/

*"Imagination is more important than knowledge."
Albert Einstein*

Research Interest

Programming languages, program analysis, verification and formal methods to provide safety and security guarantees of software.

Education

2011– **PhD, Computer Science and Automation**, *Indian Institute of Science*, Bangalore.
Ongoing, Advisor: Prof. Y. N. Srikant, jointly with Prof. Aditya Kanade.

2006–2010 **B.Tech, Computer Science and Engineering**, *Madam Mohan Malaviya Engineering College*, Gorakhpur, 7.4/10.

Experience

2012–2014 **Student System Administrator**, *Department of Computer Science and Automation*, *Indian Institute of Science*, Bangalore.

2010–2011 **Project Engineer**, *Center for Development of Advanced Computing (CDAC)*, Mumbai.
Worked in Knowledge Based Computing Group.

Skills

Programming Languages Java, C, Haskell, Agda, Gallina

Frameworks and Tools Soot, Coq, LLVM

Programming Languages Theory Type Systems, Logic, Category Theory, λ -calculus, Process Algebra (π -calculus)

Research Projects

- **Static Analysis of Android Applications** : The project designs an asynchrony-aware static analysis for Android applications. This requires a sound modeling of the control flow semantics of applications which involves modeling asynchrony, framework enforced control flow and other features which make the control flow convoluted. [MEMOCODE '16]
 - Skills acquired : Static program analysis theory and tools, like alias analysis, tpestate analysis and interprocedural analysis using graph-reachability, Soot : static analysis tool for Java, thor-

ough understanding of Android control flow and other Android internals, including asynchrony, life-cycles etc.

- **Beyond-Regular Tpestate** : The project targets the expressive limitation associated with regular tpestates. We present a generalized notion of tpestate, which we call Beyond-Regular tpestate (BR-tpestate). The BR-tpestate is a dependently typed extension of regular tpestates and thus is more expressive than regular tpestates and yet has a decidable typechecking and even a decidable type-inference in certain cases. [preprint- arXiv:1702.08154]
 - Skills acquired : Type theory, Dependent type theory, Designing a programming language syntax and semantics, Implementing a typechecker and type-inference system for a DSL, Theorem proving using Coq and familiarity with functional programming.
- **Formal modeling of Android applications using π -calculus** : The project aims at presenting a formal semantics of Android control flow using known process algebras (π -calculus). Such a semantics includes all Android control flow related intricacies, like asynchrony, event-handling callbacks and other Android framework enforced semantic features. The semantics could help proving important control flow properties over Android applications using results and tools for π -calculus. Further, the semantic could be useful for targeted code generation and optimization of Android applications.
 - Skills acquired : Process Algebra theories, like π -calculus, CCS, Actor based models etc, Android control flow semantics, Modeling systems using process algebras, Proving program properties in π -calculus.

Publications

- MEMOCODE '16** A. Mishra, A. Kanade, and Y. N. Srikant. Asynchrony-aware static analysis of android applications. In 2016 ACM/IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE), pages 163-172, Nov 2016.
- ECOOP Doctoral Symposium '17** Ashish Mishra and Y. N. Srikant. Analysis and verification of rich tpestate properties for complex programs. ECOOP 2017 Doctoral Symposium. June 18, 2017. Barcelona, Spain, 2017.
- Submitted** Ashish Mishra and Y. N. Srikant. Beyond-Regular tpestate. CoRR, abs/1702.08154, 2017.
- Ongoing** A. Mishra, Y. N. Srikant, and A. Kanade. A π -calculus Modeling of Android Control Flow Semantics and its Application to Android Static Analysis. This is a journal version of our MEMOCODE '16 paper with a formal modeling of Android control flow and soundness results of claims in the paper.

Other Projects

- **Finding null dereference bugs in Java Programs**: The project develops an intra-procedural static null dereference analysis for Java programs.
 - Skills acquired : Static program analysis theory and tools, Soot, Abstract Interpretation theory.
- **Implementation of a Code Generator for a Subset of C, for the x86-64 bit architecture** : The project implements optimization and code generation phases for a subset of C using LLVM compiler framework.

- Skills acquired : Compiler Design theory, LLVM.
- **Implementation of Scheme in Haskell** : This is a hands on project based on the examples and exercises in the book: "Write Yourself a Scheme in 48 Hours".
 - Skills acquired : Design and Implementation of a Language, Haskell, Agda.
- **Intelligent Advisory System for Farmers** : The project develops a case based reasoning system to handle farming and related queries from farmers. The projects provides a web based and mobile platform based support.
 - Skills acquired : Case based reasoning, Web applications development, Python.

Talks and Presentations

- Poster **Language Based Security for Smartphones**, Security and Privacy Symposium, 2013, Indian Institute of Technology, Kanpur, India.
- Poster **Typestate Analysis for Android Applications**, EECS Research Students Symposium, 2016, Indian Institute of Science, Bangalore, India.
- Talk **Lambda Calculus, the core of Functional Programming Languages**, Fourth Undergraduate Summer School, 2016, Indian Institute of Science, Bangalore, India.

Teaching

- Fall '16 TA for the graduate Course on **Programming Language Design and Implementation**, Indian Institute of Science.

References(More upon request)

Y. N. Srikant

Professor,
Department of Computer Science and Automation,
Indian Institute of Science, Bangalore.
email: srikant@csa.iisc.ernet.in