A Guided Tour to Static Program Analysis: State-of-the-Art Tools and Techniques

.....

Overview

Programming errors are, depending on the application, annoying till intolerable. Many theoretical methods have been developed to prove the correctness of programs. However, in practice the effort to verify realistic programs is immense, or often even unmanageable, so these methods are applied only in the most safety-critical applications and/or in special areas, for instance, hardware verification. To routinely eliminate errors in industrial programs, in contrast, only methods which identify a wide range of often encountered program errors efficiently and automatically, are acceptable.

Static analysis is a generic term for many such approaches. Examples of such approaches are:

- Identify type inconsistencies by type checking,
- Potential nil-pointer dereferencing,
- Security checks when downloading Java-applets,
- Estimations of communication overhead, and
- Resource bound checks.

This course gives the necessary background in formal methods, state-of-the art techniques in modeling and analysis of cutting edge research domains including Infotainment systems, robotics, IoT, Blockchain Smart Contracts. The substance of this course provides the scientific basis for making such systems safer, more secure and can also help them meet privacy and fairness expectations.

Objectives

The primary objectives of the course are as follows:

i) Exposing participants to the fundamentals of static analysis tools and techniques: the course gives an introduction to static analysis, emphasizing some basic techniques. The participants will learn about the principles behind static analysis and how to apply basic analysis techniques.

ii) The course will enhance the capability of the participants to identify the pros and cons of different analysis techniques and tools for program verification/certification, in particular data flow analysis and control flow analysis, inter-procedural analysis, abstract interpretation, type and effect systems.

iii) The course provides an exposure to the vulnerability issues in cutting-edge domains, such as robot operating system, blockchain smart contracts, android applications, car infotainment systems, and IoT systems and will provide evidence of the benefit of applying static analysis techniques in the software engineering life-cycle in order to prevent them and/or minimize the occurrence of the exploitation of such vulnerabilities.

Modules	**April 4 – April 15, 2022 (except Saturday & Sunday) covering 25 hours lectures and 18 hours tutorials.**
You Should Attend If…	• Executives, engineers and researchers from software industries, service and government organizations including R&D laboratories. • Student students at all levels (BTech/MSc/MTech/PhD) or Faculty from reputed academic institutions and technical institutions.
Fees	The participation fees for taking the course is as follows: **Participants from abroad : USD 200** **Industry/Research Organizations: INR 4,500** **Faculty from Academic Institutions: INR 2,500** **Indian Students: INR 1,500** The course will be conducted in online mode.

The Faculty

Prof. Agostino Cortesi is a professor of computer science with the Universita Ca' Foscari Venezia, Italy. He has extensive experience in the area of static analysis and software verification techniques. In particular, he contributes to the design and practical evaluation of abstract domains within the Abstract Interpretation framework. He coordinates the MAE Italy-India project 2017-19 "Formal Specification for Secured Software System".

Dr. Raju Halder currently working as an Assistant Professor in the Dept. of Comp. Sc. and Engg. at IIT Patna since 2012. Dr. Halder did his PhD in Computer Science from Ca' Foscari University of Venice (Italy) in 2012, and received B.Sc., B.Tech. and M.Tech. Degrees from University of Calcutta in the years 2002, 2005 and 2007 respectively. Before joining IIT Patna, he served as a post doctoral researcher with Macquarie University, Australia. He worked with the Robotics team at HASLab (University of Minho), Portugal, in 2016. Prior to his PhD, he had also worked as an associate system engineer with IBM India Pvt. Ltd. during 2007-2008. His areas of research interests include Formal Methods, Blockchain and Smart Contract, Program Analysis and Verification, Information Systems Security, etc.

Course Co-ordinator

Dr. Raju Halder
Phone: +91 612 3028009
E-mail: halder@iitp.ac.in
https://www.iitp.ac.in/~halder

Detailed Course Plan

Day	Session	Time	Topic
4 Apr 2022 (Monday)		13:30 - 14:00	**Inauguration function**
	Lecture-1	14:00 - 15:15 (AC)	Introduction, Background and Motivations; Critical Systems and Systems Dependability; Functional Vs. Non-functional Requirements;
	Tutorial-1	15:45 - 17:45 (AC)	Playing with algebraic concepts at the basis of program analysis
5 Apr 2022 (Tuesday)	Lecture-2	13:00 - 14:15 (AC)	Data flow analysis: Liveness, Reaching Definitions, Available Expressions; Control flow analysis
	Lecture-3	14:45 - 16:00 (AC)	Inter-procedural analysis, Type and Effect Systems, Algorithmic questions
	Tutorial-2	16:30 - 18:30 (AC)	An Introduction to Abstract Interpretation: Galois Connection, Fix-point Abstractions, Semantics Approximation, Decidabily Vs. Undecidability
6 Apr 2022 (Wednesday)	Lecture-4	13:00 - 14:15 (AC)	Formal Syntax and Semantics of IMP Language: Operational, Denotational and Axiomatic approaches
	Lecture-5	14:45 - 16:00 (AC)	Fix-point Semantics of while-statement; Trace Semantics; Collecting Semantics
	Tutorial-3	16:30 - 18:30 (AC)	A guided tour to static analysis: Demonstration on syntactic tool **Sonarqube** (A commercial widely used static analyzer, with a free Java scanner)
7 Apr 2022 (Thursday)	Lecture-6	13:00 - 14:15 (AC)	Non-Relational Abstract Domains: Sign, Parity, Intervals; Powerset abstract domains; Operators and Transfer Functions

	Lecture-7	14:45 - 16:00 (AC)	Relational Abstract Domains: Octagons, Polyhedra, Trapezoid Step Functions (TSF) domain; Operators and Transfer Functions
	Tutorial-4	16:30 - 18:30 (AC)	A guided tour to static analysis: Demonstration on syntactic tool ***SpotBugs*** (Free software, successor of FindBugs)
8 Apr 2022 (Friday)	Lecture-8	13:00 - 14:15 (AC)	Abstract Domains for String Values Analysis
	Lecture-9	14:45 - 16:00 (AC)	Trade-off between Precision and Efficiency of static analyses. Widening/Narrowing Operators, Reduced Product, Soundness vs. Completeness
	Tutorial-5	16:30 - 18:30 (AC)	A guided tour to static analysis: demonstration on semantics-based tool ***LiSA***
11 Apr 2022 (Monday)	Lecture-10	13:00 - 14:15 (AC)	Formalization of nonfunctional requirements as observable domains within the Abstract Interpretation Framework
	Lecture-11	14:45 - 16:00 (AC)	Managing requirement conflicts and prioritization
	Tutorial-6	16:30 - 18:30 (AC)	A guided tour to the verification of security threats: Demonstration on the tool ***GrammaTech*** CodeSonar
12 Apr 2022 (Tuesday)	Lecture-12	13:00 - 14:15 (AC)	Inter-language analyses: analysis of mobile and IoT systems
	Lecture-13	14:45 - 16:00 (AC)	Static Analysis of Database-intensive Programs: Syntax Vs. Semantics, Dependency Analysis, Database Leakage Analysis
	Tutorial-7	16:30 - 18:30 (RH)	A case study on static analysis and verification of complex Koboki Robot's ROS-based Source Codes
13 Apr 2022 (Wednesday)	Lecture-14	10:00 - 11:15 (RH)	Introduction to ROS-based Robotic Software; Open challenges related to the static analysis of ROS-based Robotic Software

	Lecture-15	11:45 - 13:00 (RH)	Introduction to Program Slicing: Static, Dynamic, Conditioned, Contract-based, etc; Slicing Refinements: Syntax Vs, Semantics; Abstract Program Slice; Database Code Slicing
	Tutorial-8	15:00 - 17:00 (RH)	Tutorials on Dependency Computation and Program Slicing
14 Apr 2022 (Thursday)	Lecture-16	10:00 - 11:15 (RH)	Introduction to Blockchain and Smart Contract, Introduction to Solidity Language, Various Safety and Security issues in Smart Contract Solidity Codes
	Lecture-17	11:45 - 13:00 (RH)	Static Smart Contract Code Analysis and Safety Verification, Smart Contract Code Optimization (including Gas Cost Optimization)
	Tutorial-9	15:00 - 17:00 (RH)	Introduction to Ethereum, Geth/Ganache/Truffle, Remix Platform, Solidity Code Writing/Compiling/Deployment
15 Apr 2022 (Friday)	Lecture-18	10:00 - 11:15 (RH)	Machine Learning in Program Analysis and Verification, Machine Learning in detecting Vulnerable Smart Contract Codes
	Lecture-19	11:45 - 13:00 (RH)	Demonstration of various Smart Contract Security and Safety Analyzers such as Oyente, SmartCheck, etc., Possible Research Directions
	Lecture-20	13:30 - 14:45 (AC)	Final discussion: open issues, research challenges, collaboration opportunities
		15:00 - 17:00	**Course Assessment Test and Certificate Distribution**

* AC: Prof. Agostino Cortesi, RH: Dr. Raju Halder