Preface

This volume contains the proceedings of the 30th edition of the International Static Analysis Symposium, SAS 2023, held on October 22–24, 2023, in Cascais, Portugal. The conference was a co-located event of SPLASH, the ACM SIGPLAN conference on Systems, Programming, Languages, and Applications: Software for Humanity.

Static analysis is widely recognized as a fundamental tool for program verification, bug detection, compiler optimization, program understanding, and software maintenance. The series of Static Analysis Symposia has served as the primary venue for the presentation of theoretical, practical, and application advances in the area. Previous symposia were held in Auckland, Chicago, Porto, Freiburg, New York, Edinburgh, Saint-Malo, Munich, Seattle, Deauville, Venice, Perpignan, Los Angeles, Valencia, Kongens Lyngby, Seoul, London, Verona, San Diego, Madrid, Paris, Santa Barbara, Venice, Pisa, Paris, Aachen, Glasgow, and Namur.

SAS 2023 called for papers on topics including, but not limited to, abstract interpretation, automated deduction, data flow analysis, debugging techniques, deductive methods, emerging applications, model checking, data science, program optimizations and transformations, program synthesis, program verification, machine learning and verification, security analysis, tool environments and architectures, theoretical frameworks, type checking, and distributed or networked systems. Besides the regular papers, the authors were encouraged to submit short submissions in the NEAT category to discuss experiences with static analysis tools, industrial reports, and case studies, along with tool papers, brief announcements of work in progress, well-motivated discussions of new questions or new areas, etc. Authors were encouraged to submit artifacts accompanying their papers to strengthen evaluations and the reproducibility of results.

The conference employed a double-blind reviewing process with an author response period, supported on EasyChair. This year, SAS had 40 full submitted papers (38 regular and two NEAT). The Program Committee used a two-round review process, where each remaining submission received at least three first-round reviews, and most four reviews, which the authors could then respond to. In addition to the PC members, 23 external reviewers were also involved in the process. The author response period was followed by Program Committee discussion where consensus was reached on the papers to be accepted, after a thorough assessment of the relevance and the quality of the work. Overall, 20 papers were accepted for publication (19 regular and one NEAT) and appear in this volume. The submitted papers were authored by researchers around the world: China, United States, France, Germany, Italy, Sweden, India, Macedonia, Taiwan, United Kingdom, Israel, Cuba, Denmark, Switzerland, Netherlands, Czechia, Japan, Mexico, and Canada.

We view the artifacts as being equally important for the success and development of static analysis as the written papers. It is important for researchers to be able to independently reproduce experiments, which is greatly facilitated by having the original artifacts available. Marc Chevalier, the artifact com-

mittee chair, set up the artifact committee. In line with SAS 2022, the authors could submit either the Docker or Virtual Machine images as artifacts. A public archival repository for the artifacts is available on Zenodo, hosted at https://zenodo.org/communities/sas-2023. The artifacts have badges awarded at three levels: Validated (correct functionality), Extensible (with source code), and Available (on the Zenodo repository). The artwork for the badges is by Arpita Biswas (Harvard University) and Suvam Mukherjee (Microsoft). SAS 2023 had 12 valid artifact submissions. The review process for the artifacts was similar to those for the papers. Each artifact was evaluated by three members of the artifact evaluation committee, and 11 out of 12 valid artifacts were accepted, at different levels.

In addition to the contributed papers, SAS 2023 also featured four invited talks by distinguished researchers: Gagandeep Singh (University of Illinois at Urbana-Champaign, VMware Research, USA), Bor-Yuh Evan Chang (U. of Colorado at Boulder), Loris D'Antoni (U. of Wiscounsin at Madison), and Daniel Kästner (AbsInt Gmbh, Germany). The Program Committee also selected the recipient of the Radhia Cousot Young Researcher Best Paper Award, given to a paper with a significant contribution from a student. This award was instituted in memory of Radhia Cousot, for her fundamental contributions to static analysis and having been one of the main promoters and organizers of the SAS series of conferences.

The SAS program would not have been possible without the efforts of many people. We thank them all. The members of the Program Committee, the artifact evaluation committee, and the external reviewers worked tirelessly to select a strong program, offering constructive and helpful feedback to the authors in their reviews. We would also like to thank the organizing committee of SPLASH 2023, chaired by Vasco T. Vasconcelos (LASIGE, University of Lisbon, Portugal) for all their efforts to make the conference a success, and the 2022 chairs Caterina Urban and Gagandeep Singh and the whole SAS Steering Committee for their help in passing the torch. We also thank our sponsors, Google, ENS Foundation, Meta, AbsInt, Springer, and the IMDEA Software Institute for their generous support of the conference. Finally, we thank Springer for publishing these proceedings.

August 8, 2023
Pozuelo de Alarcón - Madrid

Manuel V. Hermenegildo
Jose F. Morales

Table of Contents

Invited Talks

Regular Papers

Program Committee Chairs

Manuel Hermenegildo	Universidad Politécnica de Madrid and
Jose F. Morales	IMDEA Software Institute, Spain (Co-Chairs)

Steering Committee

Gagandeep Singh	VMware Research and UIUC, USA
Caterina Urban	Inria and ENS—PSL, France
Bor-Yuh Evan Chang	University of Colorado Boulder, USA
Patrick Cousot	New York University, USA
Cezara Dragoi	Inria and ENS—PSL and Informal Systems, France
Kedar Namjoshi	Nokia Bell Labs, USA
David Pichardie	Meta, France
Andreas Podelski	University of Freiburg, Germany

Program Committee

Gogul Balakrishnan	Google, USA
Liqian Chen	National University of Defense Technology, China
Yu-Fang Chen	Academia Sinica, Taiwan
Patrick Cousot	New York University, USA
Michael Emmi	Amazon Web Services, USA
Pietro Ferrara	Universitá Ca' Foscari University of Venice, Italy
Roberto Giacobazzi	University of Airzona, USA
Roberta Gori	Dipartimento di Informatica, Universita' di Pisa, Italy
Francesco Logozzo	Facebook, USA
Isabella Mastroeni	Universita' di Verona - Dipartimento di Informatica, Italy
Antoine Miné	LIP6, UPMC - Sorbonne Université, France
Kedar Namjoshi	Nokia Bell Labs, USA
Jorge A Navas	Certora Inc., USA
Martin Rinard	Massachusetts Institute of Technology, USA
Daniel Schoepe	Amazon, United Kingdom
Helmut Seidl	Technical University of Munich, Germany
Mihaela Sighireanu	LSV, ENS Paris-Saclay, France
Gagandeep Singh	University of Illinois Urbana-Champaign (UIUC) and VMware Research, USA
Fu Song	School of Information Science and Technology, ShanghaiTech University, China
Yulei Sui	University of New South Wales, Sydney, Australia
Laura Titolo	National Institute of Aerospace, NASA LaRC, USA
Jingling Xue	The University of New South Wales, Australia
Xin Zhang	Peking University, China

Artifact Evaluation Committee Chair

Marc Chevalier	Snyk, Switzerland (Chair)

Artifact Evaluation Committee

Vincenzo Arceri	University of Parma - Department of Mathematical, Physical, and Computer Sciences, Italy
Dorra Ben Khalifa	Université de Perpignan Via Domitia, France
Jérôme Boillot	École Normale Supérieure and INRIA, France
Marco Campion	INRIA & École Normale Supérieure, Université PSL, Paris, France
Yifan Chen	Peking University, China
Xiao Cheng	University of Technology, Sydney, Australia
Kai Jia	Massachusetts Institute of Technology, USA
Daniel Jurjo	IMDEA Software Institute and U. Politécnica de Madrid, Spain
Jonathan Laurent	Carnegie Mellon University / Karlsruhe Institute of Technology, USA/Germany
Denis Mazzucato	INRIA and Ecole Normale Superieure, France
Facundo Molina	IMDEA Software Institute, Spain
Luca Negrini	Corvallis SRL, Ca' Foscari University of Venice, Italy
Vivek Notani	University of Verona, Italy
Luca Olivieri	Ca' Foscari University of Venice, Italy
Francesco Parolini	Sorbonne Université, France
Louis Rustenholz	IMDEA Software Institute and U. Politécnica de Madrid, Spain
Ryan Vrecenar	Sandia National Laboratories, USA

Publicity Chair

Louis Rustenholz	IMDEA Software Institute and U. Politécnica de Madrid, Spain

Additional Reviewers

A

Arceri, Vincenzo
Ascari, Flavio
Assolini, Nicola
B

Bruni, Roberto
C

Cheng, Xiao
D

Dalla Preda, Mila
Demangeon, Romain
Dolcetti, Greta
F

Feliu Gabaldon, Marco Antonio
I

Izycheva, Anastasiia
K

Kan, Shuangxiang
L

Lei, Yuxiang
Liu, Jiaxiang
Lo, Fang-Yi
P

Petter, Michael
R

Ren, Jiawei
S

Stucki, Sandro
T

Tsai, Wei-Lun

Abstract of Invited Talk

Goal-Directed Abstract Interpretation and Event-Driven Frameworks[1]

Bor-Yuh Evan Chang[1,2] [0000−0002−1954−0774]
University of Colorado Boulder, USA and Amazon[2], USA
evan.chang@colorado.edu

Abstract. Static analysis is typically about computing a global over-approximation of a program's behavior from its source code. But what if most of the program code is missing or unknown to the analyzer? What if even where the program starts is unknown? This fundamentally thorny situation arises when attempting to analyze interactive applications (apps) developed against modern, event-driven software frameworks.

Rich event-driven software frameworks enable software engineers to create complex applications on sophisticated computing platforms (e.g., smartphones with a broad range of sensors and rich interactivity) with relatively little code by simply implementing callbacks to respond to events. But developing apps against them is also notoriously difficult. To create apps that behave as expected, developers must follow the complex and opaque asynchronous programming protocols imposed by the framework. So what makes static analysis of apps hard is essentially what makes programming them hard: the specification of the programming protocol is unclear and the possible control flow between callbacks is largely unknown.

While the typical workaround to perform static analysis with an unknown framework implementation is to either assume it to be arbitrary or attempt to eagerly specify all possible callback control flow, this solution can be too pessimistic to prove properties of interest or too burdensome and tricky to get right. In this talk, I argue for a rethinking of how to analyze app code in the context of an unknown framework implementation. In particular, I present some benefits from taking a goal-directed or backward-from-error formulation to prove just the assertions of interest and from designing semantics, program logics, specification logics, and abstract domains to reason about the app-framework boundary in a first-class manner. What follows are hopefully lines of work that make analyzing modern interactive applications more targeted, more compositional, and ultimately more trustworthy.

Keywords: goal-directed verification · backwards abstract interpretation · event-driven framework modeling

[1] I would like to especially thank the following for making significant contributions to the research described in this talk: Ph.D. students Shawn Meier, Benno Stein, and Sam Blackshear; postdoc Sergio Mover; and collaborators Manu Sridharan and Gowtham Kaki. The University of Colorado Programming Languages and Verification (CUPLV) Group has offered the essential community with insightful discussions to conduct this work. This research was supported in part by NSF awards CCF-1055066, CCF-1619282, CCF-2008369 and DARPA award FA8750-14-2-0263.

[2] Bor-Yuh Evan Chang holds concurrent appointments at the University of Colorado Boulder and as an Amazon Scholar. This talk describes work performed at the University of Colorado Boulder and is not associated with Amazon.