

38th European Conference on Object-Oriented Programming

ECOOP 2024, September 16–20, 2024, Vienna, Austria

Part 1 of 2

Edited by

Jonathan Aldrich

Guido Salvaneschi



Editors

Jonathan Aldrich 

Carnegie Mellon University, Pittsburgh, PA, USA
jonathan.aldrich@cmu.edu

Guido Salvaneschi 

University of St. Gallen, Switzerland
guido.salvaneschi@unisg.ch

ACM Classification 2012

Software and its engineering → General programming languages

ISBN 978-3-95977-341-6

PRINT ISBN: 979-8-3313-0534-5

Published online and open access by

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <https://www.dagstuhl.de/dagpub/978-3-95977-341-6>.

Publication date

September, 2024

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <https://portal.dnb.de>.

License

This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0):
<https://creativecommons.org/licenses/by/4.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.ECOOP.2024.0

ISBN 978-3-95977-341-6

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

■ Contents

Message from the Program Chairs	
<i>Jonathan Aldrich and Guido Salvaneschi</i>	0:ix
Message from the Artifact Evaluation Chairs	
<i>Karine Even-Mendoza and Raphaël Monat</i>	0:xi
Message from the AITO President	
<i>Davide Ancona</i>	0:xiii
List of Authors	
.....	0:xv

Regular Papers

A Sound Type System for Secure Currency Flow	
<i>Luca Aceto, Daniele Gorla, and Stian Lybech</i>	1:1–1:27
Runtime Instrumentation for Reactive Components	
<i>Luca Aceto, Duncan Paul Attard, Adrian Francalanza, and Anna Ingólfssdóttir</i>	2:1–2:33
A Dynamic Logic for Symbolic Execution for the Smart Contract Programming Language Michelson	
<i>Barnabas Arvay, Thi Thu Ha Doan, and Peter Thiemann</i>	3:1–3:26
Dynamically Generating Callback Summaries for Enhancing Static Analysis	
<i>Steven Arzt, Marc Miltenberger, and Julius Nümann</i>	4:1–4:27
Behavioural Up/down Casting For Statically Typed Languages	
<i>Lorenzo Bacchiani, Mario Bravetti, Marco Giunti, João Mota, and António Ravara</i>	5:1–5:28
Cross Module Quickening – The Curious Case of C Extensions	
<i>Felix Berlakovich and Stefan Brunthaler</i>	6:1–6:29
HOBBIT: Hashed OBJECT Based InTegrity	
<i>Matthias Bernad and Stefan Brunthaler</i>	7:1–7:25
Understanding Concurrency Bugs in Real-World Programs with Kotlin Coroutines	
<i>Bob Brockbernd, Nikita Koval, Arie van Deursen, and Burcu Kulahcioglu Ozkan</i> ..	8:1–8:20
A Language-Based Version Control System for Python	
<i>Luís Carvalho and João Costa Seco</i>	9:1–9:27
Indirection-Bounded Call Graph Analysis	
<i>Madhurima Chakraborty, Aakash Gnanakumar, Manu Sridharan, and Anders Møller</i>	10:1–10:22
Regrading Policies for Flexible Information Flow Control in Session-Typed Concurrency	
<i>Farzaneh Derakhshan, Stephanie Balzer, and Yue Yao</i>	11:1–11:29

38th European Conference on Object-Oriented Programming (ECOOP 2024).

Editors: Jonathan Aldrich and Guido Salvaneschi



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Mutation-Based Lifted Repair of Software Product Lines <i>Aleksandar S. Dimovski</i>	12:1–12:24
Pure Methods for roDOT <i>Vlastimil Dort, Yufeng Li, Ondřej Lhoták, and Pavel Parízek</i>	13:1–13:29
The Performance Effects of Virtual-Machine Instruction Pointer Updates <i>M. Anton Ertl and Bernd Paysan</i>	14:1–14:26
Rose: Composable Autodiff for the Interactive Web <i>Sam Estep, Wode Ni, Raven Rothkopf, and Joshua Sunshine</i>	15:1–15:27
Mover Logic: A Concurrent Program Logic for Reduction and Rely-Guarantee Reasoning <i>Cormac Flanagan and Stephen N. Freund</i>	16:1–16:29
Fair Join Pattern Matching for Actors <i>Philipp Haller, Ayman Hussein, Hernán Melgratti, Alceste Scalas, and Emilio Tuosto</i>	17:1–17:28
A CFL-Reachability Formulation of Callsite-Sensitive Pointer Analysis with Built-In On-The-Fly Call Graph Construction <i>Dongjie He, Jingbo Lu, and Jingling Xue</i>	18:1–18:29
Fearless Asynchronous Communications with Timed Multiparty Session Protocols <i>Ping Hou, Nicolas Lagaillardie, and Nobuko Yoshida</i>	19:1–19:30
Taking a Closer Look: An Outlier-Driven Approach to Compilation-Time Optimization <i>Florian Huemer, David Leopoldseder, Aleksandar Prokopec, Raphael Mosaner, and Hanspeter Mössenböck</i>	20:1–20:28
Learning Gradual Typing Performance <i>Mohammad Wahiduzzaman Khan, Sheng Chen, and Yi He</i>	21:1–21:27
CONSTRUCTOR: Immutability as a Design Concept <i>Elad Kinsbruner, Shachar Itzhaky, and Hila Peleg</i>	22:1–22:29
InferType: A Compiler Toolkit for Implementing Efficient Constraint-Based Type Inference <i>Senxi Li, Tetsuro Yamazaki, and Shigeru Chiba</i>	23:1–23:28
Qafny: A Quantum-Program Verifier <i>Liyi Li, Mingwei Zhu, Rance Cleaveland, Alexander Nicolellis, Yi Lee, Le Chang, and Xiaodi Wu</i>	24:1–24:31
Compositional Symbolic Execution for Correctness and Incorrectness Reasoning <i>Andreas Löw, Daniele Nantes-Sobrinho, Sacha-Élie Ayoun, Caroline Cronjäger, Petar Maksimović, and Philippa Gardner</i>	25:1–25:28
Matching Plans for Frame Inference in Compositional Reasoning <i>Andreas Löw, Daniele Nantes-Sobrinho, Sacha-Élie Ayoun, Petar Maksimović, and Philippa Gardner</i>	26:1–26:20
The Fault in Our Stars: Designing Reproducible Large-scale Code Analysis Experiments <i>Petr Maj, Stefanie Muroya, Konrad Siek, Luca Di Grazia, and Jan Vitek</i>	27:1–27:23

Static Basic Block Versioning <i>Olivier Melançon, Marc Feeley, and Manuel Serrano</i>	28:1–28:27
Generalizing Shape Analysis with Gradual Types <i>Zeina Migeed, James Reed, Jason Ansel, and Jens Palsberg</i>	29:1–29:28
Verifying Lock-Free Search Structure Templates <i>Nisarg Patel, Dennis Shasha, and Thomas Wies</i>	30:1–30:28
Ozone: Fully Out-of-Order Choreographies <i>Dan Plyukhin, Marco Peressotti, and Fabrizio Montesi</i>	31:1–31:28
Tenspiler: A Verified-Lifting-Based Compiler for Tensor Operations <i>Jie Qiu, Colin Cai, Sahil Bhatia, Niranjan Hasabnis, Sanjit A. Seshia, and Alvin Cheung</i>	32:1–32:28
Compiling with Arrays <i>David Richter, Timon Böhler, Pascal Weisenburger, and Mira Mezini</i>	33:1–33:24
Pipit on the Post: Proving Pre- and Post-Conditions of Reactive Systems <i>Amos Robinson and Alex Potanin</i>	34:1–34:28
Partial Redundancy Elimination in Two Iterative Data Flow Analyses <i>Reshma Roy, Sreekala S, and Vineeth Paleri</i>	35:1–35:19
Scaling Interprocedural Static Data-Flow Analysis to Large C/C++ Applications: An Experience Report <i>Fabian Schiebel, Florian Sattler, Philipp Dominik Schubert, Sven Apel, and Eric Bodden</i>	36:1–36:28
Java Bytecode Normalization for Code Similarity Analysis <i>Stefan Schott, Serena Elisa Ponta, Wolfram Fischer, Jonas Klauke, and Eric Bodden</i>	37:1–37:29
Optimizing Layout of Recursive Datatypes with Marmoset: Or, Algorithms + Data Layouts = Efficient Programs <i>Vidush Singhal, Chaitanya Koparkar, Joseph Zullo, Artem Pelenitsyn, Michael Vollmer, Mike Rainey, Ryan Newton, and Milind Kulkarni</i>	38:1–38:28
Formalizing, Mechanizing, and Verifying Class-Based Refinement Types <i>Ke Sun, Di Wang, Sheng Chen, Meng Wang, and Dan Hao</i>	39:1–39:30
Information Flow Control in Cyclic Process Networks <i>Bas van den Heuvel, Farzaneh Derakhshan, and Stephanie Balzer</i>	40:1–40:30
Refinements for Multiparty Message-Passing Protocols: Specification-Agnostic Theory and Implementation <i>Martin Vassor and Nobuko Yoshida</i>	41:1–41:29
Failure Transparency in Stateful Dataflow Systems <i>Aleksey Veresov, Jonas Spenger, Paris Carbone, and Philipp Haller</i>	42:1–42:31
Inductive Predicate Synthesis Modulo Programs <i>Scott Wesley, Maria Christakis, Jorge A. Navas, Richard Treffer, Valentin Wüstholtz, and Arie Gurfinkel</i>	43:1–43:30

Type Tailoring	
<i>Ashton Wiersdorf, Stephen Chang, Matthias Felleisen, and Ben Greenman</i>	44:1–44:27
Higher-Order Specifications for Deductive Synthesis of Programs with Pointers	
<i>David Young, Ziyi Yang, Ilya Sergey, and Alex Potanin</i>	45:1–45:26
CtChecker: A Precise, Sound and Efficient Static Analysis for Constant-Time Programming	
<i>Quan Zhou, Siruan Dang, and Danfeng Zhang</i>	46:1–46:26
Defining Name Accessibility Using Scope Graphs	
<i>Aron Zwaan and Casper Bach Poulsen</i>	47:1–47:29