

37th European Conference on Object-Oriented Programming

ECOOP 2023, July 17–21, 2023, Seattle, Washington,
United States

Edited by

Karim Ali

Guido Salvaneschi

Part 1 of 2



Editors

Karim Ali 

University of Alberta, Canada
karim.ali@ualberta.ca

Guido Salvaneschi 

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

ACM Classification 2012
Software and its engineering

ISBN 978-3-95977-281-5

PRINT ISBN: 978-1-7138-8136-0

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/978-3-95977-281-5>.

Publication date

July, 2023

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.2023.0

ISBN 978-3-95977-281-5

ISSN 1868-8969

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

■ Contents

Message from the Program Chairs <i>Karim Ali and Guido Salvaneschi</i>	0:ix
Message from the Artifact Evaluation Chairs <i>Stefan Winter and Hernan Luis Ponce de Leon</i>	0:xi
Foreword by the President of AITO <i>Eric Jul</i>	0:xiii
List of Authors	0:xv

Regular Papers

Designing Asynchronous Multiparty Protocols with Crash-Stop Failures <i>Adam D. Barwell, Ping Hou, Nobuko Yoshida, and Fangyi Zhou</i>	1:1–1:30
Nested Pure Operation-Based CRDTs <i>Jim Bauwens and Elisa Gonzalez Boix</i>	2:1–2:26
Multi-Graded Featherweight Java <i>Riccardo Bianchini, Francesco Dagnino, Paola Giannini, and Elena Zucca</i>	3:1–3:27
Hoogle \star : Constants and λ -abstractions in Petri-net-based Synthesis using Symbolic Execution <i>Henrique Botelho Guerra, João F. Ferreira, and João Costa Seco</i>	4:1–4:28
Modular Abstract Definitional Interpreters for WebAssembly <i>Katharina Brandl, Sebastian Erdweg, Sven Keidel, and Nils Hansen</i>	5:1–5:28
Dynamically Updatable Multiparty Session Protocols: Generating Concurrent Go Code from Unbounded Protocols <i>David Castro-Perez and Nobuko Yoshida</i>	6:1–6:30
Modular Compilation for Higher-Order Functional Choreographies <i>Luís Cruz-Filipe, Eva Graversen, Lovro Lugović, Fabrizio Montesi, and Marco Peressotti</i>	7:1–7:37
Wiring Circuits Is Easy as $\{0, 1, \omega\}$, or Is It... <i>Jan de Muijnck-Hughes and Wim Vanderbauwhede</i>	8:1–8:28
VeriFx: Correct Replicated Data Types for the Masses <i>Kevin De Porre, Carla Ferreira, and Elisa Gonzalez Boix</i>	9:1–9:45
On Leveraging Tests to Infer Nullable Annotations <i>Jens Dietrich, David J. Pearce, and Mahin Chandramohan</i>	10:1–10:25
super-Charging Object-Oriented Programming Through Precise Typing of Open Recursion <i>Andong Fan and Lionel Parreaux</i>	11:1–11:28

LoRe: A Programming Model for Verifiably Safe Local-First Software (Extended Abstract) <i>Julian Haas, Ragnar Mogk, Elena Yanakieva, Annette Bieniusa, and Mira Mezini</i>	12:1–12:15
Dynamic Determinacy Race Detection for Task-Parallel Programs with Promises <i>Feiyang Jin, Lechen Yu, Tiago Cogumbreiro, Jun Shirako, and Vivek Sarkar</i>	13:1–13:30
Algebraic Replicated Data Types: Programming Secure Local-First Software <i>Christian Kuessner, Ragnar Mogk, Anna-Katharina Wickert, and Mira Mezini</i> ...	14:1–14:33
Behavioural Types for Local-First Software <i>Roland Kuhn, Hernán Melgratti, and Emilio Tuosto</i>	15:1–15:28
Constraint Based Compiler Optimization for Energy Harvesting Applications <i>Yannan Li and Chao Wang</i>	16:1–16:29
Restrictable Variants: A Simple and Practical Alternative to Extensible Variants <i>Magnus Madsen, Jonathan Lindegaard Starup, and Matthew Lutze</i>	17:1–17:27
Programming with Purity Reflection: Peaceful Coexistence of Effects, Laziness, and Parallelism <i>Magnus Madsen and Jaco van de Pol</i>	18:1–18:27
Exact Separation Logic: Towards Bridging the Gap Between Verification and Bug-Finding <i>Petar Maksimović, Caroline Cronjäger, Andreas Lööw, Julian Sutherland, and Philippa Gardner</i>	19:1–19:27
Morpheus: Automated Safety Verification of Data-Dependent Parser Combinator Programs <i>Ashish Mishra and Suresh Jagannathan</i>	20:1–20:27
Automata Learning with an Incomplete Teacher <i>Mark Moeller, Thomas Wiener, Alaia Solko-Breslin, Caleb Koch, Nate Foster, and Alexandra Silva</i>	21:1–21:30
Modular Verification of State-Based CRDTs in Separation Logic <i>Abel Nieto, Arnaud Daby-Seesaram, Léon Gondelman, Amin Timany, and Lars Birkedal</i>	22:1–22:27
Information Flow Analysis for Detecting Non-Determinism in Blockchain <i>Luca Olivieri, Luca Negrini, Vincenzo Arceri, Fabio Tagliaferro, Pietro Ferrara, Agostino Cortesi, and Fausto Spoto</i>	23:1–23:25
Toward Tool-Independent Summaries for Symbolic Execution <i>Frederico Ramos, Nuno Sabino, Pedro Adão, David A. Naumann, and José Fragoso Santos</i>	24:1–24:29
A Direct-Style Effect Notation for Sequential and Parallel Programs <i>David Richter, Timon Böhler, Pascal Weisenburger, and Mira Mezini</i>	25:1–25:22
SINATRA: Stateful Instantaneous Updates for Commercial Browsers Through Multi-Version eXecution <i>Ugnius Rumsevicius, Siddhanth Venkateshwaran, Ellen Kidane, and Luís Pina</i> ...	26:1–26:29

An Efficient Vectorized Hash Table for Batch Computations <i>Hesam Shahrokhii and Amir Shaikhha</i>	27:1–27:27
Hinted Dictionaries: Efficient Functional Ordered Sets and Maps <i>Amir Shaikhha, Mahdi Ghorbani, and Hesam Shahrokhii</i>	28:1–28:30
Semantics for Noninterference with Interaction Trees <i>Lucas Silver, Paul He, Ethan Cecchetti, Andrew K. Hirsch, and Steve Zdancewic</i>	29:1–29:29
Interaction Tree Specifications: A Framework for Specifying Recursive, Effectful Computations That Supports Auto-Active Verification <i>Lucas Silver, Eddy Westbrook, Matthew Yacavone, and Ryan Scott</i>	30:1–30:26
Breaking the Negative Cycle: Exploring the Design Space of Stratification for First-Class Datalog Constraints <i>Jonathan Lindegaard Starup, Magnus Madsen, and Ondřej Lhoták</i>	31:1–31:28
Asynchronous Multiparty Session Type Implementability is Decidable – Lessons Learned from Message Sequence Charts <i>Felix Stutz</i>	32:1–32:31
ConDRust: Scalable Deterministic Concurrency from Verifiable Rust Programs <i>Felix Suchert, Lisza Zeidler, Jeronimo Castrillon, and Sebastian Ertel</i>	33:1–33:39
Dependent Merges and First-Class Environments <i>Jinhao Tan and Bruno C. d. S. Oliveira</i>	34:1–34:32
Synthesis-Aided Crash Consistency for Storage Systems <i>Jacob Van Geffen, Xi Wang, Emina Torlak, and James Bornholt</i>	35:1–35:26
Synthesizing Conjunctive Queries for Code Search <i>Chengpeng Wang, Peisen Yao, Wensheng Tang, Gang Fan, and Charles Zhang</i> ...	36:1–36:30
Do Machine Learning Models Produce TypeScript Types That Type Check? <i>Ming-Ho Yee and Arjun Guha</i>	37:1–37:28

Experience Papers

Building Code Transpilers for Domain-Specific Languages Using Program Synthesis <i>Sahil Bhatia, Sumer Kohli, Sanjit A. Seshia, and Alvin Cheung</i>	38:1–38:30
Rust for Morello: Always-On Memory Safety, Even in Unsafe Code <i>Sarah Harris, Simon Cooksey, Michael Vollmer, and Mark Batty</i>	39:1–39:27
On Using VeriFast, VerCors, Plural, and KeY to Check Object Usage <i>João Mota, Marco Giunti, and António Ravara</i>	40:1–40:29

Pearls/Brave New Ideas

The Dolorem Pattern: Growing a Language Through Compile-Time Function Execution <i>Simon Henniger and Nada Amin</i>	41:1–41:27
--	------------

0:viii **Contents**

Synthetic Behavioural Typing: Sound, Regular Multiparty Sessions via Implicit
Local Types
 Sung-Shik Jongmans and Francisco Ferreira 42:1–42:30

On the Rise of Modern Software Documentation
 Marco Raglianti, Csaba Nagy, Roberto Minelli, Bin Lin, and Michele Lanza 43:1–43:24

Python Type Hints Are Turing Complete
 Ori Roth 44:1–44:15