Proceedings of the ACM on Programming Languages
ABOUT PROCEEDINGS OF THE ACM ON PROGRAMMING LANGUAGES

Proceedings of the ACM on Programming Languages (PACMPL) is a Gold Open Access journal publishing research on all aspects of programming languages, from design to implementation and from mathematical formalisms to empirical studies. Each issue of the journal is devoted to a particular subject area within programming languages and will be announced through publicized Calls for Papers. All accepted papers receive two rounds of reviewing and authors can expect initial decisions regarding submissions in under 3 months. The journal operates in close collaboration with the Special Interest Group on Programming Languages (SIGPLAN) and is committed to making high-quality peer-reviewed scientific research in programming languages free of restrictions on both access and use.

JOURNAL WEBSITE
pacmpl.acm.org

SUBMISSIONS
More details, including instructions for submitting to PACMPL, can be found at http://pacmpl.acm.org/

Send orders to:
ACM Member Services Dept.
General Post Office
PO Box 3077
New York, NY 10087-0777

MEMBERSHIP INFORMATION
ACM Member Services Dept.
1601 Broadway, 10th Floor
New York, NY 10019-7434
T: (212) 626-0500
F: (212) 944-1318
Email: acmhelp@acm.org

PACMPL (ISSN: 2475-1421) is published quarterly in Spring, Summer, Fall, and Winter by the Association for Computing Machinery (ACM), 1601 Broadway, 10th Floor New York, NY 10019-7434

ACM COPYRIGHT NOTICE
Copyright ©2019 by the Association for Computing Machinery (ACM). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. Authors refer to the retained rights section of ACM rights forms for exceptions around posting the accepted version. The full notice is available at www.acm.org/publications/policies/copyright-policy#permanent_rights. To copy or otherwise redistribute requires prior specific permission and/or a fee. Request permission to republish from: permissions@acm.org or fax Publications Department, ACM, Inc. Fax +1 212-869-0481.
The Proceedings of the ACM series presents the highest quality research conducted in diverse areas of computer science, as represented by the ACM Special Interest Groups (SIGs). The ACM Proceedings of the ACM on Programming Languages (PACMPL) focuses on research on all aspects of programming languages, from design to implementation and from mathematical formalisms to empirical studies. The journal operates in close collaboration with the Special Interest Group on Programming Languages (SIGPLAN) and is committed to making high-quality peer-reviewed scientific research in programming languages free of restrictions on both access and use.

This issue of the PACMPL journal publishes 73 articles that were submitted in response to a call for papers seeking contributions on all aspects of programming languages and software engineering with articles targeting any stage of software development, including requirements, modeling, prototyping, design, implementation, generation, analysis, verification, testing, evaluation, maintenance, and reuse of software systems, and contributions including the development of new tools (such as language front-ends, program analyses, and runtime systems), new techniques (such as methodologies, design processes, and code organization approaches), new principles (such as formalisms, proofs, models, and paradigms), and new evaluations (such as experiments, corpora analyses, user studies, and surveys).

The articles were selected from 201 submissions — submitted by the April 2019 deadline for this issue — by means of a rigorous reviewing process. In the two-stage process, articles were evaluated with respect to the novelty and importance of their results, the evidence for these results, and the clarity of their presentation. In the first stage, each article was reviewed by at least three reviewers during a nine week review period. Additional reviews were solicited for several articles to obtain additional expert opinions. Reviews were conducted by the members of a primary review committee, a secondary review committee, and external reviewers. Authors were invited to submit a detailed response to the reviews. Based on the reviews, the author response, a one week online discussion, and a two day physical meeting of the primary review committee in Phoenix, Arizona, 10 articles were accepted with minor revisions and 63 articles required major revisions. The first stage was double blind; submissions were anonymous and the identity of authors was only revealed after the review period when that was necessary for the evaluation process, which happened only in a couple of cases. In the second stage, authors submitted non-anonymous revisions after a six week revision period with a cover letter explaining how they addressed the feedback from reviewers. Major revisions were re-reviewed by the original reviewers during a two week review period, determining whether the required revisions were satisfied. The authors of two articles were asked to make further required revisions.

I am excited by the compelling and thought-provoking work that resulted in this PACMPL issue. To provoke further discussion and dissemination, the authors were invited to also present their work to the programming languages community at the next ACM OOPSLA conference. I hope that you will also join us in October 20-25, 2019 in Athens, Greece for SPLASH/OOPSLA 2019. The conference will provide many opportunities to share ideas with programming language researchers and practitioners from institutions around the world.
It was an honor and a privilege to serve as Associate Editor for this issue of PACMPL, and I would like to thank the many people who contributed to make this a success. First, I would like to thank all the authors for contributing their work.

Second, I would like to thank the reviewers for their hard work. They have provided very useful feedback to the authors, helping them to improve their work. The high quality of the articles in this issue is also the result of their work. The Primary Review Committee consisted of Sara Achour, Nada Amin, Bor-Yuh Evan Chang, Arthur Charguéraud, Yufei Ding, Alastair Donaldson, Sebastian Erdweg, Ronald Garcia, David Grove, Görel Hedin, Martin Hirzel, Marieke Huisman, Gail Kaiser, Eric Koskinen, Ondřej Lhoták, Yu David Liu, Brandon Lucia, Heather Miller, Todd Mytkowicz, Alex Potanin, Tiark Rompf, Manu Sridharan, Friedrich Steimann, Éric Tanter, Ross Tate, Emina Torlak, David Van Horn, Eric Van Wyk, Harry Xu, Nobuko Yoshida, and Francesco Zappa Nardelli. The Secondary Review Committee consisted of Aggelos Biboudis, Gavin Bierman, Walter Binder, Eva Darulova, Werner Dietl, Isil Dillig, Sophia Drossopoulou, Susan Eisenbach, Matthew Flatt, Jeremy Gibbons, Elisa Gonzalez Boix, Sam Guyer, Christine H. Flood, Jeff Huang, Ranjit Jhala, Stephen Kell, Viktor Kuncak, Christian Kästner, Crista Lopes, Sasa Misailovic, Andrew Myers, Iulian Neamtiu, Benjamin C. Pierce, G. Ramalingam, Grigore Rosu, Malavika Samak, Jennifer B. Sartor, Peter Sewell, Xipeng Shen, Michael Steindorfer, Peter Thiemann, and Viktor Vafeiadis. The External Reviewers were Aws Albarghouthi, Timothy Bourke, Edwin Brady, David Darais, Julian Dolby, Marco Gaboardi, Rahul GoPINath, Andrew D. Gordon, Marco Guarnieri, Holger Hermanns, Felienne Hermans, Jeroen Keiren, Dan Kifer, Robbert Krebbers, Shuvendu Lahiri, Mohsen Lesani, Christof Lofi, Roman Manevich, Darya Melicher, Leo Meyerovich, Peter Müller, Bruno Oliveira, Aurojit Panda, Alexander Ratner, John Regehr, Thomas Reps, Manuel Serrano, Alexander J. Summers, Petar Tsankov, Alex Weddell, Andy Zaidman, and Hengchu Zhang.

Third I would like to thank the SPLASH 2019 conference and its General Chair, Yannis Smaragdakis, for providing the authors of this issue the opportunity to present their work.

Finally, I would like to thank the PACMPL Editorial Board and its Editor in Chief Philip Wadler for their advise, and I would like to thank SIGPLAN and its Executive Committee chaired by Jens Palsberg for supporting the gold open access publication of the articles in PACMPL and for organizing a thriving programming language community that produces high quality research as exemplified in this issue.
Contents

Editorial Message ................................................................. .................................

CLOTHO: Directed Test Generation for Weakly Consistent Database Systems
Kia Rahmani, Kartik Nagar, Benjamin Delaware, and Suresh Jagannathan — Purdue University, USA 117

Probabilistic Verification of Fairness Properties via Concentration
Osbert Bastani, Xin Zhang, and Armando Solar-Lezama — University of Pennsylvania, USA; Massachusetts Institute of Technology, USA ................................. 118

Verifying Safety and Accuracy of Approximate Parallel Programs via Canonical Sequentialization
Vimuth Fernando, Keyur Joshi, and Sasa Misailovic — University of Illinois at Urbana-Champaign, USA ................................................................. 119

Qubit Allocation as a Combination of Subgraph Isomorphism and Token Swapping
Marcos Yukio Siraichi, Vinicius Fernandes dos Santos, Caroline Collange, and Fernando Magno Quintão Pereira — Federal University of Minas Gerais, Brazil; Inria, France; University of Rennes, France; CNRS, France; IRISA, France ................................. 120

Modular Verification of Heap Reachability Properties in Separation Logic
Arshavir Ter-Gabrielyan, Alexander J. Summers, and Peter Müller — ETH Zurich, Switzerland ................................................................. 121

Complete Monitors for Gradual Types
Ben Greenman, Matthias Felleisen, and Christos Dimoulas — Northeastern University, USA; Northwestern University, USA ................................................................. 122

TLA+ Model Checking Made Symbolic
Igor Konnov, Jure Kukovec, and Thanh-Hai Tran — Inria, France; LORIA, France; University of Lorraine, France; CNRS, France; TU Vienna, Austria ................................. 123

Value-Centric Dynamic Partial Order Reduction
Krishnendu Chatterjee, Andreas Pavlovic, and Viktor Toman — IST Austria, Austria; EPFL, Switzerland ................................................................. 124

Seq: A High-Performance Language for Bioinformatics
Ariya Shajii, Ibrahim Numanagić, Riyadh Baghdadi, Bonnie Berger, and Saman Amarasinghe — Massachusetts Institute of Technology, USA ................................. 125

Staged Abstract Interpreters: Fast and Modular Whole-Program Analysis via Meta-programming
Guannan Wei, Yuxuan Chen, and Tiark Rompf — Purdue University, USA ................................................................. 126

Derivative Grammars: A Symbolic Approach to Parsing with Derivatives
Ian Henriksen, Gianfranco Bilardi, and Keshav Pingali — University of Texas at Austin, USA; University of Padua, Italy ................................................................. 127

Efficient Lock-Free Durable Sets
Yoav Zuriel, Michal Friedman, Gali Sheffi, Nachshon Cohen, and Erez Petrank — Technion, Israel; Amazon, Israel ................................................................. 128

Modular Verification for Almost-Sure Termination of Probabilistic Programs
Mingzhang Huang, Hongfei Fu, Krishnendu Chatterjee, and Amir Kafshdar Goharshady — Shanghai Jiao Tong University, China; East China Normal University, China; IST Austria, Austria ................................................................. 129

IVT: An Efficient Method for Sharing Subtype Polymorphic Objects
Yu-Ping Wang, Xu-Qiang Hu, Zi-Xin Zou, Wende Tan, and Gang Tan — Tsinghua University, China; Pennsylvania State University, USA ................................................................. 130

Refinement Kinds: Type-Safe Programming with Practical Type-Level Computation
Luís Caires and Bernardo Toninho — Nova University of Lisbon, Portugal; NOVA-LINCS, Portugal ................................................................. 131
Program Synthesis with Algebraic Library Specifications  
Benjamin Mariano, Josh Reese, Siyuan Xu, ThanhVu Nguyen, Xiaokang Qiu, Jeffrey S. Foster, and Armando Solar-Lezama — University of Maryland at College Park, USA; Purdue University, USA; University of Nebraska-Lincoln, USA; Tufts University, USA; Massachusetts Institute of Technology, USA

Weakening WebAssembly  
Conrad Watt, Andreas Rossberg, and Jean Pichon-Pharabod — University of Cambridge, UK; Dfinity Stiftung, Germany

Generating a Fluent API with Syntax Checking from an LR Grammar  
Tetsuro Yamazaki, Tomoki Nakamaru, Kazuhiro Ichikawa, and Shigeru Chiba — University of Tokyo, Japan

Weak Persistency Semantics from the Ground Up: Formalising the Persistency Semantics of ARMv8 and Transactional Models  
Azalea Raad, John Wickerson, and Viktor Vafeiadis — MPI-SWS, Germany; Imperial College London, UK

DeepSEA: A Language for Certified System Software  
Vilhelm Sjöberg, Yuyang Sang, Shu-chun Weng, and Zhong Shao — Yale University, USA; CertiK, USA

BDA: Practical Dependence Analysis for Binary Executables by Unbiased Whole-Program Path Sampling and Per-Path Abstract Interpretation  
Zhuo Zhang, Wei You, Guanhong Tao, Guannan Wei, Yonghwi Kwon, and Xiangyu Zhang — Purdue University, USA; Renmin University of China, China; University of Virginia, USA

Asphalion: Trustworthy Shielding against Byzantine Faults  
Ivana Vukotic, Vincent Rahli, and Paulo Esteves-Verissimo — University of Luxembourg, Luxembourg; University of Birmingham, UK

Automatic Repair of Regular Expressions  
Rong Pan, Qinheping Hu, Gaowei Xu, and Loris D’Antoni — University of Texas at Austin, USA; University of Wisconsin-Madison, USA

Static Analysis with Demand-Driven Value Refinement  
Benno Stein, Benjamin Barslev Nielsen, Bor-Yuh Evan Chang, and Anders Møller — University of Colorado Boulder, USA; Aarhus University, Denmark

Relational Verification using Reinforcement Learning  
Jia Chen, Jiayi Wei, Yu Feng, Osbert Bastani, and Isil Dillig — University of Texas at Austin, USA; University of California at Santa Barbara, USA; University of Pennsylvania, USA

A Formalization of Java’s Concurrent Access Modes  
John Bender and Jens Palsberg — University of California at Los Angeles, USA

On the Fly Synthesis of Edit Suggestions  
Anders Miltner, Sumit Gulwani, Vu Le, Alan Leung, Arjun Radhakrishna, Gustavo Soares, Ashish Tiwari, and Abhishek Udupa — Princeton University, USA; Microsoft, USA

A Fault-Tolerant Programming Model for Distributed Interactive Applications  
Ragnar Mogk, Joscha Drechsler, Guido Salvaneschi, and Mira Mezini — TU Darmstadt, Germany

A Path to DOT: Formalizing Fully Path-Dependent Types  
Marianna Rapoport and Ondřej Lhoták — University of Waterloo, Canada

Reliable and Fast DWARF-Based Stack Unwinding  
Théophile Bastian, Stephen Kell, and Francesco Zappa Nardelli — ENS, France; University of Kent, UK; Inria, France

Leveraging Rust Types for Modular Specification and Verification  
Vytautas Astrauskas, Peter Müller, Federico Poli, and Alexander J. Summers — ETH Zurich, Switzerland
Precision-Preserving Yet Fast Object-Sensitive Pointer Analysis with Partial Context Sensitivity
Jingbo Lu and Jingling Xue — UNSW, Australia .......................... 148

Formal Foundations of Serverless Computing
Abhinav Jangda, Donald Pinckney, Yuriy Brun, and Arjun Guha — University of Massachusetts Amherst, USA .......................... 149

Optimal Stateless Model Checking for Reads-From Equivalence under Sequential Consistency
Parosh Aziz Abdulla, Mohamed Faouzi Atig, Bengt Jonsson, Magnus Lång, Tuan Phong Ngo, and Konstantinos Sagonas — Uppsala University, Sweden .......................... 150

Modular Verification of Web Page Layout
Pavel Panchekha, Michael D. Ernst, Zachary Tatlock, and Shoaib Kamil — University of Utah, USA; University of Washington, USA; Adobe, USA .......................... 151

Aroma: Code Recommendation via Structural Code Search
Sifei Luan, Di Yang, Celeste Barnaby, Koushik Sen, and Satish Chandra — Facebook, USA; University of California at Irvine, USA; University of California at Berkeley, USA .......................... 152

On the Design, Implementation, and Use of Laziness in R
Aviral Goel and Jan Vitek — Northeastern University, USA; Czech Technical University, Czechia .......................... 153

Mergeable Replicated Data Types
Gowtham Kaki, Swarn Priya, KC Sivaramakrishnan, and Suresh Jagannathan — Purdue University, USA; IIT Madras, India .......................... 154

Compiler Fuzzing: How Much Does It Matter?
Michaël Marcozzi, Qiyi Tang, Alastair F. Donaldson, and Cristian Cadar — Imperial College London, UK .......................... 155

DProf: Distributed Profiler with Strong Guarantees
Zachary Benavides, Keval Vora, and Rajiv Gupta — University of California at Riverside, USA; Simon Fraser University, Canada .......................... 156

Precise Reasoning with Structured Time, Structured Heaps, and Collective Operations
Grégory M. Essertel, Guannan Wei, and Tiark Rompf — Purdue University, USA .......................... 157

Casting about in the Dark: An Empirical Study of Cast Operations in Java Programs
Luis Mastrangelo, Matthias Hauswirth, and Nathaniel Nystrom — USI Lugano, Switzerland .......................... 158

Getafix: Learning to Fix Bugs Automatically
Johannes Bader, Andrew Scott, Michael Pradel, and Satish Chandra — Facebook, USA .......................... 159

Generating Precise Error Specifications for C: A Zero Shot Learning Approach
Baijun Wu, John Peter Campora III, Yi He, Alexander Schlecht, and Sheng Chen — University of Louisiana at Lafayette, USA .......................... 160

Specifying Concurrent Programs in Separation Logic: Morphisms and Simulations
Aleksandar Nanosvki, Anindya Banerjee, Germán Andrés Delbianco, and Ignacio Fábregas — IMDEA Software Institute, Spain; IRIF, France; University of Paris, France .......................... 161

Improving Bug Detection via Context-Based Code Representation Learning and Attention-Based Neural Networks
Yi Li, Shaohua Wang, Tien N. Nguyen, and Son Van Nguyen — New Jersey Institute of Technology, USA; University of Texas at Dallas, USA .......................... 162

Scala Implicits Are Everywhere: A Large-Scale Study of the Use of Scala Implicits in the Wild
Filip Křikava, Heather Miller, and Jan Vitek — Czech Technical University, Czechia; Carnegie Mellon University, USA; Northeastern University, USA .......................... 163

Optimization of Swift Protocols
Rajkishore Barik, Manu Sridharan, Murali Krishna Ramanathan, and Milind Chabbi — Uber Technologies, USA; University of California at Riverside, USA .......................... 164
On the Complexity of Checking Transactional Consistency
Ranadeep Biswas and Constantin Enea — University of Paris, France; IRIF, France; CNRS, France

System FR: Formalized Foundations for the Stainless Verifier
Jad Hamza, Nicolas Voirol, and Viktor Kunčak — EPFL, Switzerland

Language-Integrated Privacy-Aware Distributed Queries
Guido Salvaneschi, Mirko Köhler, Daniel Sokolowski, Philipp Haller, Sebastian Erdweg, and Mira Mezini — TU Darmstadt, Germany; KTH, Sweden; Johannes Gutenberg University Mainz, Germany

AutoPandas: Neural-Backed Generators for Program Synthesis
Rohan Bavishi, Caroline Lemieux, Roy Fox, Koushik Sen, and Ion Stoica — University of California at Berkeley, USA; University of California at Irvine, USA

Ryū Revisited: Printf Floating Point Conversion
Ulf Adams — Google, Germany

IntelliMerge: A Refactoring-Aware Software Merging Technique
Bo Shen, Wei Zhang, Haiyan Zhao, Guangtai Liang, Zhi Jin, and Qianxiang Wang — Peking University, China; Huawei Technologies, China

Certifying Graph-Manipulating C Programs via Localizations within Data Structures
Shengyi Wang, Qinxiang Cao, Anshuman Mohan, and Aquinas Hobor — National University of Singapore, Singapore; Shanghai Jiao Tong University, China

Duet: An Expressive Higher-Order Language and Linear Type System for Statically Enforcing Differential Privacy
Joseph P. Near, David Darais, Chike Abua, Tim Stevens, Pranav Gaddamadugu, Lun Wang, Neel Somani, Mu Zhang, Nikhil Sharma, Alex Shan, and Dawn Song — University of Vermont, USA; University of California at Berkeley, USA; University of Utah, USA

Effective Lock Handling in Stateless Model Checking
Michalis Kokologiannakis, Azalea Raad, and Viktor Vafeiadis — MPI-SWS, Germany

FuzzFactory: Domain-Specific Fuzzing with Waypoints
Rohan Padhye, Caroline Lemieux, Koushik Sen, Laurent Simon, and Hayawardh Vijayakumar — University of California at Berkeley, USA; Samsung Research, USA

AL: Autogenerating Supervised Learning Programs
José P. Cambronero and Martin C. Rinard — Massachusetts Institute of Technology, USA

Sound and Reusable Components for Abstract Interpretation
Sven Keidel and Sebastian Erdweg — Johannes Gutenberg University Mainz, Germany

Design, Implementation, and Application of GPU-Based Java Bytecode Interpreters
Ahmet Celik, Pengyu Nie, Christopher J. Rossbach, and Milos Gligoric — University of Texas at Austin, USA; VMware, USA

Specification and Inference of Trace Refinement Relations
Timos Antonopoulos, Eric Koskinen, and Ton Chanh Le — Yale University, USA; Stevens Institute of Technology, USA

Dependence-Aware, Unbounded Sound Predictive Race Detection
Kaan Genç, Jake Roemer, Yufan Xu, and Michael D. Bond — Ohio State University, USA

Trace Aware Random Testing for Distributed Systems
Burcu Kulahcioğlu Özkan, Rupak Majumdar, and Simin Oraee — MPI-SWS, Germany

Coverage Guided, Property Based Testing
Leonidas Lampropoulos, Michael Hicks, and Benjamin C. Pierce — University of Maryland, USA; University of Pennsylvania, USA

PlanAlyzer: Assessing Threats to the Validity of Online Experiments
Emma Tosch, Eytan Bakshy, Emery D. Berger, David D. Jensen, and J. Eliot B. Moss — University of Massachusetts Amherst, USA; Facebook, USA
I/O Dependent Idempotence Bugs in Intermittent Systems
Milijana Surbatovich, Limin Jia, and Brandon Lucia — Carnegie Mellon University, USA

Initialize Once, Start Fast: Application Initialization at Build Time
Christian Wimmer, Codrut Stancu, Peter Hofer, Vojin Jovanovic, Paul Wögerer, Peter B. Kessler, Oleg Pliss, and Thomas Würthinger — Oracle Labs, USA; Oracle Labs, Austria; Oracle Labs, Switzerland

Safer Smart Contract Programming with Scilla
Ilya Sergey, Vaivaswatha Nagaraj, Jacob Johannsen, Amrit Kumar, Anton Trunov, and Ken Chan Guan Hao — Yale-NUS College, Singapore; National University of Singapore, Singapore; Zilliqa Research, India; Zilliqa Research, Denmark; Zilliqa Research, UK; Zilliqa Research, Russia; Zilliqa Research, Malaysia

ApproxHPVM: A Portable Compiler IR for Accuracy-Aware Optimizations
Hashim Sharif, Prakalp Srivastava, Muhammad Huzaifa, Maria Kotsifakou, Keyur Joshi, Yasmin Sarita, Nathan Zhao, Vikram S. Adve, Sasa Misailovic, and Sarita Adve — University of Illinois at Urbana-Champaign, USA; Cornell University, USA

Reflection-Aware Static Regression Test Selection
August Shi, Milica Hadzi-Tanovic, Lingming Zhang, Darko Marinov, and Owolabi Legunsen — University of Illinois at Urbana-Champaign, USA; University of Texas at Dallas, USA

Automatic and Scalable Detection of Logical Errors in Functional Programming Assignments
Dowon Song, Myungho Lee, and Hakjoo Oh — Korea University, South Korea

Detecting Nondeterministic Payment Bugs in Ethereum Smart Contracts
Shuai Wang, Chengyu Zhang, and Zhendong Su — Hong Kong University of Science and Technology, China; East China Normal University, China; ETH Zurich, Switzerland

Author Index