

22nd ACM SIGPLAN International Conference on Functional Programming

Oxford, 3rd to 9th September 2017





Maths Institute Floorplan



Wifi: Connect to eduroam, or sign up (for free) to The Cloud.

Contents

Preface	page 2
Committees	page 4
Keynotes	page 6
Evening Events	page 7
Workshop on Higher-Order Programming with EffectsProgramming Languages Mentoring WorkshopScheme and Functional Programming WorkshopWorkshop on Type-Driven DevelopmentICFP Tutorials	page 8 page 9 page 10 page 11 page 12
ICFP – Day 1	page 13
FSCD – Day 1	page 14
ICFP – Day 2	page 15
FSCD – Day 2	page 16
ICFP – Day 3	page 17
FSCD – Day 3	page 18
FSCD – Day 4	page 19
Haskell Symposium – Day 1	page 20
ML Family Workshop	page 21
Workshop on Functional High-Performance Computing	page 22
CUFP Tutorials – Day 1	page 23
Haskell Symposium – Day 2	page 24
OCaml Users and Developers Workshop	page 25
Erlang Workshop	page 26
CUFP Tutorials – Day 2	page 27
Commercial Users of Functional Programming	page 28
Haskell Implementors Workshop	page 29
Functional Art, Music, Modeling and Design	page 30
In memoriam Takayuki Muranushi (1983–2017)	page 31

Message from the Chairs

Welcome to Oxford, and to ICFP 2017, the 22nd ACM SIGPLAN International Conference on Functional Programming!

ICFP provides a forum for sharing and discussing the latest work on the art and science of functional programming. We hope that you will enjoy the conference, including not only the technical presentations on a wide spectrum of topics, but also the opportunities for learning from and interacting with many researchers, developers, and students from around the world.

This year's call for papers resulted in a near-record of 127 submissions, including 20 functional pearls and 4 experience reports. From these, the program committee selected 44 papers for presentation at the conference, including 6 pearls and 2 experience reports. In addition, the technical program includes invited keynotes by Chris Martens and Rich Hickey.

Following recent practice, ICFP 2017 employed a lightweight double-blind reviewing process, with each paper receiving three or more reviews from members of the program committee and a pool of external reviewers. Initial reviews were made available to authors, many of whom provided useful feedback and clarification during a 72-hour author response period. This was followed by online discussion and an in-person meeting of the program committee in Portland, Oregon, where the final selection of papers was made. Papers whose author list included a member of the program committee were not discussed until all other decisions had been made, and were held to a higher standard; ultimately, 4 of the 12 papers in this category were accepted.

All of the papers accepted for ICFP 2017 are being published in the inaugural edition of a new journal, Proceedings of the ACM on Programming Languages (PACMPL), which 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.

One of the more visible changes resulting from our participation in PACMPL is the new, singlecolumn document format. The transition from the previous two-column format was not always easy, either for authors or reviewers. We thank everyone for their patience and goodwill throughout multiple iterations of this process, and we hope that the community will benefit from resulting improvements in readability and accessibility.

A less obvious impact of the move to PACMPL was the introduction of a new, two-phase selection and reviewing process, further enhancing the already rigorous approach that had been used in prior years. Concretely, this allowed for papers to be "conditionally accepted" at the end of the program committee meeting, and for the associated reviews to be annotated with a list of "mandatory revisions." Authors of conditionally accepted papers then had approximately five weeks to revise and submit their papers for a second and final reviewing phase, including a cover letter to explain how the mandatory revisions had been addressed. In total, 30 papers were conditionally accepted, with mandatory revisions ranging from minor to more substantial but all judged to be feasible given the limited five week period for revision. While it is difficult to quantify the overall impact of this process, there were many cases where the extended opportunities for interaction between authors and reviewers helped to improve quality in both technical details and presentation. Ultimately, all conditionally accepted papers were accepted for inclusion in the conference.

Another new feature for ICFP this year that will be apparent from the papers in the proceedings is the introduction of an artifact evaluation process, with selected papers receiving a corresponding "badge" or "seal of approval" on their first page. Artifact evaluation, which supports future researchers in reproducing and building on current work, has proved to be a valuable component of other conferences, and we are confident that the ICFP community will benefit in a similar way. Although the process is optional, all authors of accepted papers were invited to prepare and submit artifacts to accompany and support their paper. These items were reviewed by an artifact evaluation committee that also provided feedback to authors to help improve the quality of submitted artifacts. The committee received 31 artifact submissions, all of which were accepted, and 27 of the associated papers are further badged as having a publicly available artifact.

As usual, the main conference is complemented by a range of affiliated events, including twelve co-hosted conferences, workshops or symposia, as well as the ICFP Programming Contest and the Student Research Competition, with results for both announced during the conference. In addition to technically focussed workshops on a broad range of topics, we are proud to include the SIGPLAN *Programming Languages Mentoring Workshop* (PLMW) at ICFP. The purpose of this mentoring workshop is to encourage senior undergraduate and beginning graduate students to pursue careers in programming language research, and to engage them in a process of imagining how they might contribute to the world. A novelty this year is that ICFP will run in parallel with FSCD 2017, the *Second International Conference on Formal Structures for Computation and Deduction*, with joint lunch and coffee breaks and the same overall schedule structure so that registrants at either conference can attend talks at the other.

It is hard to overstate the fundamental importance of community to the good health and success of ICFP. Of course, this includes the authors and developers that share their work, and the attendees who provide a stimulating environment for discussion and debate. But special recognition is due for the many volunteers who-even when they are already very busy with other commitments-still step up to take on new responsibilities and roles in support of the conference. We are deeply humbled, impressed, and grateful for their commitment, hard work, and expertise. In particular, we would like to acknowledge the program committee, the external reviewers, and the artifact evaluation committee for their thorough and thoughtful reviews; Annabel Satin and Marta Zampollo, for their excellent arrangements; the members of the ICFP steering committee, for their long term stewardship and dedication to the success of the conference; Ryan Trinkle, for liaising with our industrial partners and sponsors; David Christiansen and Andres Löh for their leadership in organizing the associated workshops; Neelakantan Krishnaswami, Dan Licata, and Brigitte Pientka for chairing PLMW; Ilya Sergey for running the Student Research Competition; Sam Lindley for coordinating the ICFP Programming Contest; Lindsey Kuper for making sure that the ICFP community is informed and engaged; José Calderón for managing the process of recording and posting videos for many of the talks at ICFP and associated events; Dirk Beyer and Conference Publishing Consulting for compiling the proceedings; the PACMPL Editorial Board and the staff at ACM headquarters—especially Philip Wadler, Michael Hicks, Matthew Fluet, Scott Delman, Craig Rodkin, and Laura Lander-for guiding and supporting the transition to PACMPL; Ted Cooper and Larry Diehl for their assistance during the program committee meeting; and last, but not least, Yosuke Fukuda, Yuki Nishida, and Jakub Zalewski for organizing and leading the team of student volunteers that will be working to keep everything running smoothly during the conference.

We are indebted to our partners who made it possible to keep the cost of registration reasonable, and who provided support for students who would not have been able to attend without financial aid. We are also very grateful for the generous support of ACM and ACM SIGPLAN, including their commitment to PACMPL's Gold Open Access policy, making high-quality, peer-reviewed scientific research available without restrictions on access or (re-)use. The generosity of our supporters is key in helping our community to grow and thrive.

Welcome again to ICFP. We hope you enjoy the conference, the affiliated events, the location, and the opportunities that it all provides to meet new people and, for those who have attended previously, to reconnect with colleagues and friends!

Jeremy Gibbons, University of Oxford, UK ICFP 2017 General Chair

Mark P. Jones, Portland State University, USA ICFP 2017 Program Chair

Ryan R. Newton, Indiana University, USA Matthew Flatt, University of Utah, USA ICFP 2017 Artifact Evaluation Committee Co-Chairs

Committees

Organizing Committee

General Chair: Jeremy Gibbons Programme Chair: Mark P. Jones Artifact Evaluation Chairs: Matthew Flatt, Ryan R. Newton PLMW Chairs: Neelakantan R. Krishnaswami, Dan Licata, Brigitte Pientka Workshops Chairs: David Christiansen, Andres Löh Programming Contest Organiser: Sam Lindley Video Chair: José Calderón Student Research Competition Chair: Ilya Sergey Student Volunteer Captains: Yosuke Fukuda, Yuki Nishida, Jakub Zalewski Industrial Relations Chair: Ryan Trinkle Publicity and Web Chair: Lindsey Kuper

Program Committee

Robert Atkey Adam Chlipala **Dominique Devriese** Martin Erwig Matthew Flatt **Ronald Garcia** Kathryn E. Gray John Hughes Chung-Kil Hur Graham Hutton Alan Jeffrey Ranjit Ihala Shin-ya Katsumata Lindsey Kuper Dan Licata **Ben Lippmeier** Gabriel Scherer Alexandra Silva Nikhil Swamy Sam Tobin-Hochstadt Nicolas Wu Beta Ziliani

Additional Reviewers

José Calderón David Christiansen Ted Cooper Joshua Dunfield Matthew Fluet Adrien Guatto Martin Hofmann Oleg Kiselyov

University of Strathclyde, UK Massachusetts Institute of Technology, USA KU Leuven, Belgium Oregon State University, USA University of Utah, USA University of British Columbia, Canada University of Cambridge, and Facebook Inc., UK Chalmers University of Technology, Sweden Seoul National University, South Korea University of Nottingham, UK Mozilla Research, USA University of California at San Diego, USA National Institute of Informatics, Japan Intel Labs, USA Wesleyan University, USA Digital Asset / UNSW, Australia Northeastern University, USA University College London, UK Microsoft Research, USA Indiana University, USA University of Bristol, UK FAMAF, UNC and CONICET, Argentina

Daan Leijen Gopalan Nadathur Daniel Patterson Mike Rainey Tiark Rompf Daniel Roy John Reppy David Sands Justin Slepak Matthieu Sozeau David Thibodeau Meng Wang Jeremy Yallop Brent Yorgey

Artifact Evaluation Committee

Michael Ballantyne Matt Brown Buddhika Chamith Jesper Cockx Erik de Castro Lopo Alejandro Gadea Jason Gross **Emmanuel Gunther** Eric Holk Kuen-Bang Hou (Favonia) **Timothy Jones** Georgios Karachalias Wen Kokke Nicholas Labich Trevor L. McDonell Joshua Moerman Phúc C. Nguyễn Matthew Pickering Tahina Ramananandro Ryan Scott Mallku Ernesto Soldevila Raffa Jacob Stanley Michel Steuwer **Bo** Joel Svensson Michael Vollmer Timothy A. K. Zakian

ICFP Steering Committee

Derek Dreyer **Robert Bruce Findler** Kathleen Fisher Matthew Flatt **Jacques** Garrigue Jeremy Gibbons Michael Hicks John Hughes Atsushi Igarashi Mark Jones Gabriele Keller Lindsey Kuper John Reppy Michael Sperber Eijiro Sumii Wouter Swierstra Peter Thiemann

University of Utah, USA UCLA, USA Indiana University, USA KU Leuven, Belgium Ambiata, Australia FAMAF, UNC and CONICET, Argentina MIT CSAIL, USA FAMAF, UNC and CONICET, Argentina Google, USA Carnegie Mellon University, USA Victoria University of Wellington, New Zealand KU Leuven, Belgium University of Edinburgh, UK University of Maryland, USA University of New South Wales, Australia Radboud University, Netherlands University of Maryland, USA University of Bristol, UK Microsoft Research, USA Indiana University, USA Universidad Nacional de Córdoba and CONICET, Argentina Ambiata, Australia University of Edinburgh, UK Chalmers University of Technology, Sweden Indiana University, USA University of Oxford, UK

MPI-SWS, Germany Northwestern University, USA Tufts University, USA University of Utah, USA Nagoya University, Japan University of Oxford, UK University of Maryland, USA Chalmers University of Technology, Sweden Kyoto University, Japan Portland State University, USA Data61, CSIRO (formerly NICTA) and UNSW, Australia Intel Labs, USA University of Chicago, USA Active Group GmbH, Germany Tohoku University, Japan University of Utrecht, Netherlands University of Freiburg, Germany

ICFP Keynotes

Compositional Creativity: Some Principles for Talking to Computers

Chris Martens (North Carolina State University)

Abstract: *Generativity* is an increasingly popular and useful concept, referring to a machine's ability to respond to user input with new constructions not foreseen by the programmer. Yet increasingly, people treat computational systems as unknowable black-box systems, writing off the possibility of forming mental models that allow a collaborative relationship between human insight and fast computation.

I argue for the efficacy of transparent, compositional semantics for collaborating with virtual agents and deriving insights from system models. Having built systems based on automated reasoning for linear logic and epistemic modal logic, we can formalize notions of belief, intention, and action, in order to create virtual agents that behave in ways that humans can reason about based on intuitions about goal-driven behavior. For example, some of Grice's maxims of conversation can be seen as derivable consequences of these principles. Ongoing work includes applying these formalisms to the tasks of navigating unknown rule systems in virtual environments, social skills training, and generative storytelling.

Bio: Chris Martens is an Assistant Professor at North Carolina State University, where she directs the Principles of Expressive Machines (POEM) lab. She was a postdoctoral researcher with UC Santa Cruz's Expressive Intelligence Studio after completing her Ph.D. with the Principles of Programming group at Carnegie Mellon in 2015. At CMU she worked on logical frameworks, dependent type theory, and representing narrative structure with linear logic. Her current research activities include applying automated reasoning and compositional semantics to the authorship of generative systems, games, and narrative, as well as designing accessible tools for system modeling and logical specification.

Ten Years of Clojure—FP Out of the Box

Rich Hickey (Cognitect Inc.)

Abstract: Since its first release in 2007, Clojure has seen considerable adoption in industries as diverse as Fortune 5 retail, international banking and finance, climate science and the latest startups. Clojure has tens of thousands of users and is a top 20 language in the RedMonk 2017 index. This is somewhat unexpected (especially by its author!) given that Clojure is a Lisp dialect emphasizing functional programming. What happened?

This talk examines the objectives of Clojure and its initial and ongoing design decisions. Moreover, we can now examine how that's worked out—which features are valued by its users and how it is employed to build systems. We will briefly touch on Datomic, a functional database system written in Clojure by the same authors that completes the model of programming Clojure espouses. Finally we'll look at what challenges remain for programmers and how Clojure might grow to address them.

Bio: Rich Hickey is the author of Clojure, and the designer of Datomic. Rich has 30 years of professional experience in all facets of software development. Rich has worked on scheduling systems, broadcast automation, audio analysis and fingerprinting, database systems, yield management, exit poll systems, and machine listening, in a variety of languages.

Social Events

• Monday 4th: Welcome Reception

18:30-20:30, in the Foyer in the Maths Institute

• Wednesday 6th: Lambda Ladies Lunch

12:00–13:00, in the Foyer in the Maths Institute

• Wednesday 6th: Banquet

18:30-22:30, in the Blackwell Hall of the Weston Library, on Broad Street

• Thursday 7th: Industry Reception

18:15-20:30, in the Ashmolean Museum, on Beaumont Street

• Saturday 9th: FARM Evening of Algorithmic Arts

19:30-22:00, in the Old Fire Station, on George Street

(For a location map, see the inside back cover.)

09:00 - 09:10 **Welcome**

09:00 Welcome

François Pottier and Aleksandar Nanevski

09:10 - 10:10 Invited talk

09:10 Invited Talk: Semantics of Effect Systems by Graded Monads Shin-ya Katsumata

10:30 – 11:30 Modular Semantics

- 10:30 Higher-order Programming is an Effect Oleg Kiselyov
- 11:00 A monadic solution to the Cartwright-Felleisen-Wadler conjecture Ohad Kammar and Dylan McDermott

12:00 - 12:30 **Rust**

12:00 RustBelt: Securing the Foundations of the Rust Programming Language Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers and Derek Dreyer

14:00 – 15:00 Effects and Dependent Types

- 14:00 Handling fibred algebraic effects Danel Ahman
- 14:30 Only Control Effects and Dependent Types Youyou Cong and William J. Bowman

15:30 – 16:30 Effects

- 15:30 Programming a Web Server with Algebraic Effects Daan Leijen
- 16:00 Logical Relations for Algebraic Effects Dariusz Biernacki, Maciej Piróg, Piotr Polesiuk and Filip Sieczkowski

16:50 – 17:20 **Monotonicity**

16:50 Recalling a Witness Danel Ahman, Cătălin Hriţcu, Kenji Maillard, Aseem Rastogi, Nikhil Swamy and Cédric Fournet

Programming Languages Mentoring Workshop

09:00 – 09:10 **Welcome**

09:00 Welcome

Brigitte Pientka, Neelakantan R. Krishnaswami and Dan Licata

09:10 - 10:00 **Keynote**

09:10 Keynote Chris Martens

10:30 – 11:30 **Session 1**

- 10:30 A Few Frank Remarks Conor McBride
- 11:00 Compositional Compiler Correctness Amal Ahmed

12:00 – 12:30 Session 2

12:00 TBD

14:00 - 15:00 **Session 3**

14:00 Gradual Typing *Ronald Garcia*14:30 Scala: Types in Theory & Practice *Nada Amin*

15:30 - 16:30 Session 4

15:30 How to Write Papers and Give Talks That People Can Follow Derek Dreyer

16:50 - 17:40 Session 5

16:50 Panel Discussion: Careers in Programming Languages Sam Staton and Richard A. Eisenberg

Scheme and Functional Programming Workshop

09:00 - 09:10 Session 0

09:00 Welcome

09:10 - 10:10 Session 1

09:10 Keynote Sam Tobin-Hochstadt

10:30 – 11:30 **Session 2**

- 10:30 Paper: Scalar and Tensor Parameters for Importing Tensor Index Notation including Einstein Summation Notation Satoshi Egi
- 11:15 Lightning Talk: Extending the LISP model from cons cells to triples, from trees to hypergraphs Joe Corneli and Raymond Puzio

12:00 – 12:30 Session 3

12:00 Panel: Future of Scheme François-René Rideau, Marc Feeley, Arthur Gleckler, Kathy Gray, Alaric Snell-Pym and Andy Wingo

14:00 - 15:00 Session 4

- 14:00 Paper: Toward Parallelizing Control-flow Analysis with Datalog Thomas Gilray and Sidharth Kumar
- 14:45 Lightning: Gerbil on Gambit, as they say Racket on Chez Dimitris Vyzovitis

15:30 – 16:30 **Session 5**

- 15:30 Report: Status of the ongoing R7RS standardization process *Alaric Snell-Pym*
- 16:15 Lightning: lambda talk Alain Marty

16:50 – 17:50 Session 6

- 16:50 Invited Talk Matthew Might
- 17:20 Goodbye

Workshop on Type-Driven Development

09:00 – 09:05 **Welcome**

09:00 Welcome Sam Lindley and Brent Yorgey

09:10 – 10:00 **Invited talk**

09:10 Driving types into PHP Andrew Kennedy

10:30 – 11:30 Full papers 1

- 10:30 Type-directed diffing of structured data Victor Cacciari Miraldo, Pierre-Evariste Dagand and Wouter Swierstra
- 11:00 Structured asynchrony with algebraic effects Daan Leijen

12:00 – 12:25 Extended abstracts 1

12:00 Cogent[↑]: giving systems engineers a stepping stone *Zilin Chen*

14:00 – 15:00 **Full papers 2**

- 14:00 Type safe Redis queries a case study of type-level programming in Haskell *Ting-Yan Lai, Tyng-Ruey Chuang and Shin-Cheng Mu*
- 14:30 Generic packet descriptions: verified parsing and pretty printing of low-level data Marcell van Geest and Wouter Swierstra

15:30 – 16:20 **Extended abstracts 2**

15:30 Affine killing Kiko Fernandez-Reyes and Dave Clarke
15:55 On ringads and foldables James McKinna

16:50 – 17:40 Extended abstracts 3

- 16:50 Type oriented programming for task based parallelism Nick Brown, Ludovic Capelli and James Mark Bull
- 17:15 Type-directed reasoning for probabilistic, non-compositional resources Edwin Brady, Kevin Hammond and Christopher Schwaab

ICFP Tutorials

09:10 - 12:30	Morning tutorial
09:10-10:00	
10:30-11:30	
12:00-12:30	Tutorial T1: Writing Verified Programs in CakeML
	Ramana Kumar, Michael Norrish, Scott Owens and Magnus O. Myreen
14:00 - 17:50	Afternoon tutorial
14:00-15:00	
15:00-16:20	
16:50-17:50	Tutorial T2: Certified Functional (Co)programming with Isabelle/HOL

ICFP – Day I

09:00 -	10:00 Monday Keynote	Chair: Jeremy Gibbons
09:00 09:10	Welcome Compositional creativity: some principles for talking to computers <i>Chris Martens</i>	
10:30 -	12:00 Art and Education	Chair: Kathryn Gray
10:30	Super 8 Languages for Making Movies (Functional Pearl) [#30] Leif Andersen, Stephen Chang and Matthias Felleisen	
10:52	Testing and Debugging Functional Reactive Programming [#2] Ivan Perez and Henrik Nilsson	
11:15	Lock-Step Simulation Is Child's Play (Experience Report) [#3] Joachim Breitner and Chris Smith	
11:37	Scaling up Functional Programming Education [#4] Benjamin Canou, Roberto Di Cosmo and Grégoire Henry	
13:00 -	14:30 Functional Programming Techniques	Chair: Graham Hutton
13:00	Faster Coroutine Pipelines [#5] Mike Spivey	
13:22	A Pretty But Not Greedy Printer (Functional Pearl) [#6] Jean-Philippe Bernardy	
13:45	Generic Functional Parallel Algorithms: Scan and FFT [#7] Conal Elliott	
14:07	A Unified Approach to Solving Seven Programming Problems (Fun	ctional Pearl) [#8]
	William E. Byra, Michael Ballantyne, Gregory Rosenblatt and Matthew	Might
15:00 -	16:10 Applications	<i>Might</i> Chair: Alexandra Silva
15:00 – 15:00	 William E. Byra, Michael Ballantyne, Gregory Rosenblatt and Matthew 16:10 Applications Prototyping a Query Compiler using Coq (Experience Report) [#9] Joshua Auerbach, Martin Hirzel, Louis Mandel, Avraham Shinnar and 	Might Chair: Alexandra Silva] Jerome Simeon
15:00 – 15:00 15:23	 William E. Byrd, Michael Ballantyne, Gregory Rosenblatt and Matthew 16:10 Applications Prototyping a Query Compiler using Coq (Experience Report) [#9] Joshua Auerbach, Martin Hirzel, Louis Mandel, Avraham Shinnar and A Framework for Adaptive Differential Privacy [#10] Daniel Winograd-Cort, Andreas Haeberlen, Aaron Roth and Benjamin (Composition) 	Might Chair: Alexandra Silva] Jerome Simeon C. Pierce
15:00 – 15:00 15:23 15:46	 William E. Byra, Michael Ballantyne, Gregory Rosenblatt and Matthew 1 16:10 Applications Prototyping a Query Compiler using Coq (Experience Report) [#9] Joshua Auerbach, Martin Hirzel, Louis Mandel, Avraham Shinnar and A Framework for Adaptive Differential Privacy [#10] Daniel Winograd-Cort, Andreas Haeberlen, Aaron Roth and Benjamin O Symbolic Conditioning of Arrays in Probabilistic Programs [#11] Praveen Narayanan and Chung-chieh Shan 	Might Chair: Alexandra Silva] Jerome Simeon C. Pierce
15:00 – 15:00 15:23 15:46 16:40 –	 William E. Byra, Michael Ballantyne, Gregory Rosenblatt and Matthew 1 16:10 Applications Prototyping a Query Compiler using Coq (Experience Report) [#9] Joshua Auerbach, Martin Hirzel, Louis Mandel, Avraham Shinnar and A Framework for Adaptive Differential Privacy [#10] Daniel Winograd-Cort, Andreas Haeberlen, Aaron Roth and Benjamin O Symbolic Conditioning of Arrays in Probabilistic Programs [#11] Praveen Narayanan and Chung-chieh Shan 18:10 Effects 	Might Chair: Alexandra Silva Jerome Simeon C. Pierce Chair: Ben Lippmeier
15:00 - 15:00 15:23 15:46 16:40 - 16:40	 William E. Byra, Michael Ballantyne, Gregory Rosenblatt and Matthew 1 16:10 Applications Prototyping a Query Compiler using Coq (Experience Report) [#9] Joshua Auerbach, Martin Hirzel, Louis Mandel, Avraham Shinnar and A Framework for Adaptive Differential Privacy [#10] Daniel Winograd-Cort, Andreas Haeberlen, Aaron Roth and Benjamin O Symbolic Conditioning of Arrays in Probabilistic Programs [#11] Praveen Narayanan and Chung-chieh Shan 18:10 Effects Abstracting Definitional Interpreters [#12] David Darais, Nicholas Labich, Phúc C. Nguyen and David Van Horn 	Might Chair: Alexandra Silva] Jerome Simeon C. Pierce Chair: Ben Lippmeier
15:00 - 15:00 15:23 15:46 16:40 - 16:40 17:02	 William E. Byra, Michael Ballantyne, Gregory Rosenblatt and Matthew 16:10 Applications Prototyping a Query Compiler using Coq (Experience Report) [#9] Joshua Auerbach, Martin Hirzel, Louis Mandel, Avraham Shinnar and A Framework for Adaptive Differential Privacy [#10] Daniel Winograd-Cort, Andreas Haeberlen, Aaron Roth and Benjamin (Symbolic Conditioning of Arrays in Probabilistic Programs [#11] Praveen Narayanan and Chung-chieh Shan 18:10 Effects Abstracting Definitional Interpreters [#12] David Darais, Nicholas Labich, Phúc C. Nguyen and David Van Horn On the Expressive Power of User-Defined Effects [#13] Yannick Forster, Ohad Kammar, Sam Lindley and Matija Pretnar 	Might Chair: Alexandra Silva] Jerome Simeon C. Pierce Chair: Ben Lippmeier
15:00 - 15:00 15:23 15:46 16:40 - 16:40 17:02 17:25	 William E. Byrd, Michael Ballantyne, Gregory Rosenblatt and Matthew 16:10 Applications Prototyping a Query Compiler using Coq (Experience Report) [#99 Joshua Auerbach, Martin Hirzel, Louis Mandel, Avraham Shinnar and A Framework for Adaptive Differential Privacy [#10] Daniel Winograd-Cort, Andreas Haeberlen, Aaron Roth and Benjamin O Symbolic Conditioning of Arrays in Probabilistic Programs [#11] Praveen Narayanan and Chung-chieh Shan 18:10 Effects Abstracting Definitional Interpreters [#12] David Darais, Nicholas Labich, Phúc C. Nguyen and David Van Horn On the Expressive Power of User-Defined Effects [#13] Yannick Forster, Ohad Kammar, Sam Lindley and Matija Pretnar Imperative Functional Programs That Explain Their Work [#14] Wilmer Ricciotti, Jan Stolarek, Roly Perera and James Cheney 	Might Chair: Alexandra Silva] Jerome Simeon C. Pierce Chair: Ben Lippmeier
15:00 - 15:00 15:23 15:46 16:40 - 16:40 17:02 17:25 17:47	 Witham E. Byra, Michael Battantyne, Gregory Rosenblatt and Matthew 16:10 Applications Prototyping a Query Compiler using Coq (Experience Report) [#9] Joshua Auerbach, Martin Hirzel, Louis Mandel, Avraham Shinnar and A Framework for Adaptive Differential Privacy [#10] Daniel Winograd-Cort, Andreas Haeberlen, Aaron Roth and Benjamin (Symbolic Conditioning of Arrays in Probabilistic Programs [#11] Praveen Narayanan and Chung-chieh Shan 18:10 Effects Abstracting Definitional Interpreters [#12] David Darais, Nicholas Labich, Phúc C. Nguyen and David Van Horn On the Expressive Power of User-Defined Effects [#13] Yannick Forster, Ohad Kammar, Sam Lindley and Matija Pretnar Imperative Functional Programs That Explain Their Work [#14] Wilmer Ricciotti, Jan Stolarek, Roly Perera and James Cheney Effect-Driven QuickChecking of Compilers [#15] Jan Midtgaard, Mathias Nygaard Justesen, Patrick Kasting, Flemming N Nielson 	Might Chair: Alexandra Silva] Jerome Simeon C. Pierce Chair: Ben Lippmeier

18:10 Monday Announcements

FSCD – Day I

10:30 – 10:45 **Welcome**

10:30 Welcome message Sam Staton and Dale Miller

10:45 – 11:45 **Session 1**

10:45 Brzozowski Goes Concurrent – A Kleene Theorem for Pomset Languages Alexandra Silva

13:00 – 14:30 Session 2

- 13:00 Polynomial running times for polynomial-time oracle machines Akitoshi Kawamura and Florian Steinberg
- 13:30 A Curry-Howard Approach to Church's Synthesis Colin Riba and Pierre Pradic
- 14:00 Streett Automata Model Checking of Higher-Order Recursion Schemes Ryota Suzuki, Koichi Fujima, Naoki Kobayashi and Takeshi Tsukada

15:00 – 16:00 Session 3

- 15:00 Relating System F and $\lambda 2$: A Case Study in Coq, Abella and Beluga Jonas Kaiser, Brigitte Pientka and Gert Smolka
- 15:30 Nested Multisets, Hereditary Multisets, and Syntactic Ordinals in Isabelle/HOL Jasmin Blanchette, Mathias Fleury and Dmitriy Traytel

16:40 – 18:10 Session 4

- 16:40 A polynomial-time algorithm for the Lambek calculus with brackets of bounded order Max Kanovich, Stepan Kuznetsov, Glyn Morrill and Andre Scedrov
- 17:10 A sequent calculus for semi-associativity Noam Zeilberger
- 17:40 Combinatorial Flows and their Normalisation Lutz Strassburger

ICFP – Day 2

09:00 -	10:00	Tuesday Keynote	Chair: Mark Jones
09:00	Ten Ye <i>Rich</i>	ears of Clojure – FP Out of the Box <i>Hickey</i>	
10:30 -	12:00	Low-level and Systems Programming	Chair: Sam Tobin-Hochstadt
10:30	Persist	tence for the Masses: RRB-Vectors in a Systems Languag	ge [#16]
10:52	Juan Verifie Jona	ed Low-Level Programming Embedded in F* [#17] than Protzenko et al.	
11:15	Verify Scott	ing Efficient Function Calls in CakeML [#18] Owens, Michael Norrish, Ramana Kumar, Magnus O. Myree	rn, Yong Kiam Tan
11:37	Better Geof	frey Mainland	
13:00 -	14:30	Foundations of Higher-Order Programming	Chair: Gabriel Scherer
13:00	Found <i>Thib</i>	lations of Strong Call by Need [#20] aut Balabonski, Pablo Barenbaum, Eduardo Bonelli and Delia	a Kesner
13:22	A Rela Aleja	ntional Logic for Higher-Order Programs [#21] andro Aguirre, Gilles Barthe, Marco Gaboardi, Deepak Garg a	nd Pierre-Yves Strub
13:45	How t braic ⁷ Mak	o Prove Your Calculus Is Decidable: Practical Application Theories and Computation [#22] <i>oto Hamana</i>	ons of Second-Order Alge-
14:07	No-Br Milo	cainer CPS Conversion [#23] Davis, William Meehan and Olin Shivers	
15:00 -	16:10	Tools for Verification	Chair: Nikhil Swamy
15:00	Kami: Verific Joon	A Platform for High-Level Parametric Hardware Spe cation [#24] won Choi, Muralidaran Vijayaraghavan, Benjamin Sherman,	cification and Its Modular Adam Chlipala and Arvind
15:23	Space Kons Zach	Junea Search: A Library for Building and Verifying Solver-Aid- tantin Weitz, Steven Lyubomirsky, Stefan Heule, Emina Ton ary Tatlock	ed Tools [#25] rlak, Michael D. Ernst and
15:46	Local Benj	Refinement Typing [#26] amin Cosman and Ranjit Jhala	
16:40 -	17:50	Program Construction	Chair: John Hughes
16:40	Comp Cond	iling to Categories [#27] al Elliott	
17:03	Visitor <i>Fran</i>	rs Unchained [#28] çois Pottier	
17:26	Stageo Jeren	d Generic Programming [#29] ny Yallop	
17:50 -	18:20	Tuesday Closing Events	
17:50	Progra Sam	amming Contest Report <i>Lindley</i>	

FSCD – Day 2

10:30 Uniform Resource Analysis by Rewriting: Strenghts and Weaknesses *Georg Moser*

11:30 – 12:00 Session 6

11:30 Continuation Passing Style for Effect Handlers Daniel Hillerström, Sam Lindley, Bob Atkey and KC Sivaramakrishnan

13:00 – 14:30 Session 7

13:00	Confluence of an extension of Combinatory Logic by Boolean constants	5
	Lukasz Czajka	

- 13:30 Improving Rewriting Induction Approach for Proving Ground Confluence Takahito Aoto, Yoshihito Toyama and Yuta Kimura
- 14:00 The confluent terminating context-free substitutive rewriting system for the λ -calculus with surjective pairing and terminal type *Yohji Akama*

15:00 – 16:00 **Session 8**

- 15:00 Is the optimal implementation inefficient? Elementarily not Stefano Guerrini and Marco Solieri
- 15:30 Optimality and the Linear Substitution Calculus Pablo Barenbaum and Eduardo Bonelli

16:40 – 18:10 **Session 9**

16:40	Generalized Refocusing: from Hybrid Strategies to Abstract Machines
	Malgorzata Biernacka, Witold Charatonik and Klara Zielińska
17:10	Observably Deterministic Concurrent Strategies and Intensional Full Abstraction for
	Parallel-or
	Simon Castellan, Pierre Clairambault and Glynn Winskel

17:40 Refutation of Sallé's Longstanding Conjecture Benedetto Intrigila, Giulio Manzonetto and Andrew Polonsky

ICFP – Day 3

09:00 -	10:00 SRC Presentations and Domain-Specific Languages	Chair: Martin Erwig
09:00 09:37	Student Research Competition: Finalist Presentations Herbarium Racketensis: A Stroll through the Woods (Functional Pe Vincent St-Amour, Daniel Feltey, Spencer P. Florence, Shu-Hung You and	arl) [#1] l Robert Bruce Findler
10:30 -	12:00 Dependently Typed Programming	Chair: Dan Licata
10:30 10:52	A Specification for Dependent Types in Haskell [#31] Stephanie Weirich, Antoine Voizard, Pedro Avezedo and Richard Eisenber Parametric Quantifiers for Dependent Type Theory [#32] Andreas Nuyts, Andrea Vezzosi and Dominique Devriese	g
11:15	Normalization by Evaluation for Sized Dependent Types [#33] Andreas Abel_Andrea Verzosi and Theo Winterhalter	
11:37	A Metaprogramming Framework for Formal Verification [#34] Gabriel Ebner, Sebastian Ullrich, Jared Roesch, Jeremy Avigad and Leon	ardo De Moura
13:00 -	14:30 Contracts and Sessions	Chair: Matthew Flatt
13:00	Chaperone Contracts for Higher-Order Sessions [#35] Hernan Melgratti and Luca Padovani	
13:22	Whip: Higher-Order Contracts for Modern Services [#36]	
13:45	Manifest Sharing with Session Types [#37] Stephanie Balzer and Frank Pfenning	
14:07	Gradual Session Types [#38] Atsushi Igarashi, Peter Thiemann, Vasco Vasconcelos and Philip Wadler	
15:00 -		
	16:10 Integrating Static and Dynamic Typing	Chair: Ron Garcia
15:00	 16:10 Integrating Static and Dynamic Typing Theorems for Free for Free: Parametricity, With and Without Types Amal Ahmed, Dustin Jamner, Jeremy G. Siek and Philip Wadler 	Chair: Ron Garcia [#39]
15:00 15:23	 16:10 Integrating Static and Dynamic Typing Theorems for Free for Free: Parametricity, With and Without Types Amal Ahmed, Dustin Jamner, Jeremy G. Siek and Philip Wadler On Polymorphic Gradual Typing [#40] Yuu Igarashi, Taro Sekiyama and Atsushi Igarashi 	Chair: Ron Garcia [#39]
15:00 15:23 15:46	 16:10 Integrating Static and Dynamic Typing Theorems for Free for Free: Parametricity, With and Without Types Amal Ahmed, Dustin Jamner, Jeremy G. Siek and Philip Wadler On Polymorphic Gradual Typing [#40] Yuu Igarashi, Taro Sekiyama and Atsushi Igarashi Gradual Typing with Union and Intersection Types [#41] Giuseppe Castagna and Victor Lanvin 	Chair: Ron Garcia [#39]
15:00 15:23 15:46 16:40 -	 16:10 Integrating Static and Dynamic Typing Theorems for Free for Free: Parametricity, With and Without Types Amal Ahmed, Dustin Jamner, Jeremy G. Siek and Philip Wadler On Polymorphic Gradual Typing [#40] Yuu Igarashi, Taro Sekiyama and Atsushi Igarashi Gradual Typing with Union and Intersection Types [#41] Giuseppe Castagna and Victor Lanvin 17:50 Inference and Analysis 	Chair: Ron Garcia [#39] Chair: Mark Jones
15:00 15:23 15:46 16:40 - 16:40	 16:10 Integrating Static and Dynamic Typing Theorems for Free for Free: Parametricity, With and Without Types Amal Ahmed, Dustin Jamner, Jeremy G. Siek and Philip Wadler On Polymorphic Gradual Typing [#40] Yuu Igarashi, Taro Sekiyama and Atsushi Igarashi Gradual Typing with Union and Intersection Types [#41] Giuseppe Castagna and Victor Lanvin 17:50 Inference and Analysis Constrained Type Families [#42] J. Garrett Morris and Richard A. Eisenberg 	Chair: Ron Garcia [#39] Chair: Mark Jones
15:00 15:23 15:46 16:40 - 16:40 17:03	 16:10 Integrating Static and Dynamic Typing Theorems for Free for Free: Parametricity, With and Without Types Amal Ahmed, Dustin Jamner, Jeremy G. Siek and Philip Wadler On Polymorphic Gradual Typing [#40] Yuu Igarashi, Taro Sekiyama and Atsushi Igarashi Gradual Typing with Union and Intersection Types [#41] Giuseppe Castagna and Victor Lanvin 17:50 Inference and Analysis Constrained Type Families [#42] J. Garrett Morris and Richard A. Eisenberg Automating Sized-Type Inference for Complexity Analysis [#43] Martin Avanzini and Ugo Dal Lago 	Chair: Ron Garcia [#39] Chair: Mark Jones
15:00 15:23 15:46 16:40 - 16:40 17:03 17:26	 16:10 Integrating Static and Dynamic Typing Theorems for Free for Free: Parametricity, With and Without Types <i>Amal Ahmed, Dustin Jamner, Jeremy G. Siek and Philip Wadler</i> On Polymorphic Gradual Typing [#40] <i>Yuu Igarashi, Taro Sekiyama and Atsushi Igarashi</i> Gradual Typing with Union and Intersection Types [#41] <i>Giuseppe Castagna and Victor Lanvin</i> 17:50 Inference and Analysis Constrained Type Families [#42] <i>J. Garrett Morris and Richard A. Eisenberg</i> Automating Sized-Type Inference for Complexity Analysis [#43] <i>Martin Avanzini and Ugo Dal Lago</i> Inferring Scope through Syntactic Sugar [#44] <i>Justin Pombrio, Shriram Krishnamurthi and Mitchell Wand</i> 	Chair: Ron Garcia [#39] Chair: Mark Jones
15:00 15:23 15:46 16:40 - 16:40 17:03 17:26	 16:10 Integrating Static and Dynamic Typing Theorems for Free for Free: Parametricity, With and Without Types Amal Ahmed, Dustin Jamner, Jeremy G. Siek and Philip Wadler On Polymorphic Gradual Typing [#40] Yuu Igarashi, Taro Sekiyama and Atsushi Igarashi Gradual Typing with Union and Intersection Types [#41] Giuseppe Castagna and Victor Lanvin 17:50 Inference and Analysis Constrained Type Families [#42] J. Garrett Morris and Richard A. Eisenberg Automating Sized-Type Inference for Complexity Analysis [#43] Martin Avanzini and Ugo Dal Lago Inferring Scope through Syntactic Sugar [#44] Justin Pombrio, Shriram Krishnamurthi and Mitchell Wand 18:20 Wednesday Closing Events 	Chair: Ron Garcia [#39] Chair: Mark Jones

18:10 ICFP 2018 Announcement

FSCD – Day 3

09:10 Quantitative semantics for probabilistic programming *Christine Tasson*

10:30 – 12:00 Session 11

- 10:30 Displayed categories Benedikt Ahrens and Peter Lefanu Lumsdaine
 11:00 List Objects with Algebraic Structure
 - Marcelo Fiore and Philip Saville
- 11:30 There is only one notion of differentiation Robin Cockett and Jean-Simon Lemay

13:00 – 14:30 Session 12

- 13:00 A Fibrational Framework for Substructural and Modal Logics Dan Licata, Michael Shulman and Mitchell Riley
- 13:30 Dinaturality between syntax and semantics *Paolo Pistone*
- 14:00 Models of Type Theory Based on Moore Paths Andrew Pitts and Ian Orton

15:00 - 16:00 Session 13

- 15:00 Böhm Reduction in Infinitary Term Graph Rewriting Systems Patrick Bahr
- 15:30 Infinite Runs in Abstract Completion Nao Hirokawa, aart Middeldorp, Christian Sternagel and Sarah Winkler

16:40 – 17:10 Session 14

16:40 Negative Translations and Normal Modality Tadeusz Litak, Miriam Polzer and Ulrich Rabenstein

17:10 – 17:20 Termination and Complexity Competition 2017

17:10 Termination and Complexity Competition 2017 Jürgen Giesl, Albert Rubio, Johannes Waldmann and Akihisa Yamada

17:20 – 18:10 FSCD General Meeting

17:20 FSCD General Meeting

FSCD – Day 4

09:10 -	10:00	Session	15
~ ~ ~ ~ ~			

09:10 Type systems for the relational verification of higher order programs Marco Gaboardi

10:30 – 11:59 **Session 16**

10:30	Arrays and References in Resource Aware ML
	Benjamin Lichtman and Jan Hoffmann
11 00	

- 11:00 The Complexity of Principal Inhabitation Andrej Dudenhefner and Jakob Rehof
- 11:30 Types as Resources for Classical Natural Deduction Delia Kesner and Pierre Vial

Haskell Symposium – Day I

L1

09:10 - 10:00 **Day 1, Session 1**

Ornaments: exploiting parametricity for safer, more automated code refactorization and 09:10 code reuse (Invited Talk) Didier Rémy

10:30 - 11:30 Day 1, Session 2

- 10:30 Algebraic Graphs with Class (Functional Pearl) Andrey Mokhov
- 11:00 Packrats Parse in Packs Mario Blažević and Jacques Légaré

12:00 - 12:30Day 1, Session 3

12:00 Ode on a Random Urn (Functional Pearl) Leonidas Lampropoulos, Antal Spector-Zabusky and Kenneth Foner

14:00 – 15:00 **Day 1, Session 4**

QuickSpec: A Lightweight Theory Exploration Tool for Programmers (System Demon-14:00stration)

Maximilian Algehed, Koen Claessen, Moa Johansson and Nicholas Smallbone

14:30 Speculate: Discovering Conditional Equations and Inequalities about Black-Box Functions by Reasoning from Test Results Rudy Braquehais and Colin Runciman

15:30 - 16:30 Day 1, Session 5

- 15:30Using Coq to Write Fast and Correct Haskell John Wiegley and Benjamin Delaware
- 16:00 A Tale of Two Provers: Verifying Monoidal String Matching in Liquid Haskell and Coq Niki Vazou, Leonidas Lampropoulos and Jeff Polakow

16:50 - 17:50Day 1, Session 6

- 16:50A Meta-EDSL for Distributed Web Applications Anton Ekblad
- 17:20**Composable Network Stacks and Remote Monads** Justin Dawson, Mark Grebe and Andy Gill

ML Family Workshop

09:00 - 09:00	9:05 Welcome
09:00 V	Welcome
09:10 - 1	0:00 Invited talk
09:10 S	State machines all the way down Edwin Brady
10:30 – 1	1:45 Types and modules
10:30 N	Mergeable types Gowtham Kaki, KC Sivaramakrishnan, Samodya Abeysiriwardane and Suresh Jagannathan
10:55	Terless modules Gabriel Radanne and Jérôme Vouillon
11:20 I	First-class subtypes Jeremy Yallop and Stephen Dolan
12:00 - 12	2:25 Verification
12:00 V	VOCAL – a verified OCAml Library Arthur Charguéraud, Jean-Christophe Filliatre, Mário Pereira and François Pottier
14:00 - 1	5:15 Programming language design
14:00	Typer: an infix statically typed Lisp Pierre Delaunay, Vincent Archambault-Bouffard and Stefan Monnier
14:25 I	Relational conversion for OCaml Petr Lozov and Dmitri Boulytchev
14:50	Towards abductive functional programming Koko Muroya
15:30 - 1	6:20 Performance
15:30 N	Making SML# a general-purpose high-performance language Atsushi Ohori, Kenjiro Taura and Katsuhiro Ueno
15:55 I	Efficient representation of large, dynamic sequences in ML Arthur Charguéraud and Mike Rainey
16:50 – 1	7:40 Effects
16:50 I	Effects without monads: non-determinism Oleg Kiselyov
17:15 I	Effectively tackling the awkward squad Stephen Dolan, Spiros Eliopoulos, Daniel Hillerström, Anil Madhavapeddy, KC Sivaramakr- ishnan and Leo White

Workshop on Functional High-Performance Computing

09:00 – 09:10 Welcome to FHPC'17

09:00 Welcome to FHPC'17 Phil Trinder and Cosmin Oancea

09:10 – 10:10 First Keynote

09:10 Haskell in the Datacentre Simon Marlow

10:30 – 11:30 **Compilation**

- 10:30 From High-level Radio Protocol Specifications to Efficient Low-level Implementations via Partial Evaluation Geoffrey Mainland and Siddhanathan Shanmugam
- 11:00 Destination-Passing Style for Efficient Memory Management Amir Shaikhha, Andrew Fitzgibbon, Simon Peyton Jones and Dimitrios Vytiniotis

12:00 - 12:30 **Tools**

12:00 VisPar: Visualising dataflow graphs from the Par monad Maximilian Algehed and Patrik Jansson

14:00 – 15:00 Parallel Programming

14:00 In Search of a Map: using Program Slicing to Discover Potential Parallelism in Recursive Functions

Adam Barwell and Kevin Hammond

14:30 Strategies for Regular Segmented Reductions on GPU Rasmus Wriedt Larsen and Troels Henriksen

15:30 – 16:30 **Demo Session**

15:30 Futhark Demo Troels Henriksen
16:00 Paraformance Demo Christopher Brown undefined

16:50 – 17:50 **Demo and Panel**

- 16:50 Ziria Demo Siddhanathan Shanmugam and Geoffrey Mainland
 17:90 Demol Discussion
- 17:20 Panel Discussion

CUFP Tutorials – Day I

09:10 - 12:30	CUFP Tutorials C1	L5
09:10-10:00 10:30-11:30 12:00-12:30	Tutorial C1: Online Applications with Incremental Yaron Minsky and Sebastian Funk	
09:10 - 12:30	CUFP Tutorials C2	L6
09:10-10:00 10:30-11:30 12:00-12:30	Tutorial C2: Extensible Effects: understanding them, implementing them, usi <i>Oleg Kiselyov</i>	ng the
14:00 - 17:30	CUFP Tutorials C3	L5
14:00–15:00 15:30–16:20 16:50–17:30	Tutorial C3: Concurrent Programming with Effect Handlers Daniel Hillerström and KC Sivaramakrishnan	
14:00 - 17:30	CUFP Tutorials C4	L6
14:00–15:00 15:30–16:20 16:50–17:30	Tutorial C4: Git under the hood with OCaml Romain Calascibetta	

23

Haskell Symposium – Day 2

L1

09:10 - 10:00 Day 2, Session 1

09:10 Algorithmic Music in Haskell (Invited Talk) Donya Quick

10:30 – 11:30 **Day 2, Session 2**

- 10:30 Well-Typed Music Does Not Sound Wrong (Experience Report) Dmitrij Szamozvancev and Michael Gale
- 11:00 Back to the Future: Time Travel in FRP Ivan Perez

12:00 – 12:30 **Day 2, Session 3**

12:00 The Linearity Monad Jennifer Paykin and Steve Zdancewic

14:00 – 15:00 **Day 2, Session 4**

- 14:00 Elaboration on Functional Dependencies Georgios Karachalias and Tom Schrijvers
- 14:30 Quantified Class Constraints Gert-Jan Bottu, Georgios Karachalias, Tom Schrijvers, Bruno C. d. S. Oliveira and Philip Wadler

15:30 – 16:30 **Day 2, Session 5**

15:30 Hardware Software Co-Design in Haskell Markus Aronsson and Mary Sheeran
16:00 Streaming Irregular Arrays Robert Clifton-Everest, Trevor L. McDonell, Manuel Chakravarty and Gabriele Keller

16:50 – 17:50 Day 2, Session 6

- 16:50 Improving STM Performance with Transactional Structs Ryan Yates and Michael Scott
 17:20 Adaptive Lock-Free Data Structures in Haskell: A General Method for Concurrent Im-
- plementation Swapping Chao-Hong Chen, Vikraman Choudhury and Ryan R. Newton

OCaml Users and Developers Workshop

00.00	
09.00	Opening Cabriel Scherer
09:10	Invited talk: new contributors
00.40	David Allsopp, Florian Angeletti and Sébastien Hinderer
09:40	Anil Madhavapeddy
10:30 -	11:30 Talk session 2
10:30	Owl: A General-Purpose Numerical Library in OCaml Liang Wang
10:50	Extending OCaml's open
11:10	Runhang Li and Jeremy Yallop Genspio: Generating Shell Phrases In OCaml
	Sebastien Mondet
11:35 -	12:30 Poster session
٠	mSAT: An OCaml SAT Solver (Bury Guillaume)
•	Jouilder: a modern approach to OCaml development (Jeremie Dimino, Mark Shinwell) Tyre – Typed Regular Expressions (Gabriel Radanne)
•	ocamli: Interpreted OCaml (John Whitington)
14:00 -	15:00 Talk session 3
14:00	ROTOR: First Steps Towards a Refactoring Tool for OCaml
14:20	A memory model for multicore OCaml
14.40	Stephen Dolan and KC Sivaramakrishnan
14:40	Stephen Dolan and KC Sivaramakrishnan Bioinformatics, The Typed Tagless Final Way Sebastien Mondet
14:40	Stephen Dolan and KC Sivaramakrishnan Bioinformatics, The Typed Tagless Final Way Sebastien Mondet
14:40 15:30 -	Stephen Dolan and KC Sivaramakrishnan Bioinformatics, The Typed Tagless Final Way Sebastien Mondet 16:30 Talk session 4
14:40 15:30 – 15:30	Stephen Dolan and KC Sivaramakrishnan Bioinformatics, The Typed Tagless Final Way Sebastien Mondet 16:30 Talk session 4 A B-tree library for OCaml Tom Bidge
14:40 15:30 – 15:30 15:50	Stephen Dolan and KC Sivaramakrishnan Bioinformatics, The Typed Tagless Final Way Sebastien Mondet 16:30 Talk session 4 A B-tree library for OCaml Tom Ridge Wodan: a pure OCaml, flash-aware filesystem library
14:40 15:30 - 15:30 15:50	Stephen Dolan and KC Sivaramakrishnan Bioinformatics, The Typed Tagless Final Way Sebastien Mondet 16:30 Talk session 4 A B-tree library for OCaml Tom Ridge Wodan: a pure OCaml, flash-aware filesystem library Gabriel de Perthuis Tenen the OCempl Compton Lorente Ledware
14:40 15:30 - 15:30 15:50 16:10	 Stephen Dolan and KC Sivaramakrishnan Bioinformatics, The Typed Tagless Final Way Sebastien Mondet 16:30 Talk session 4 A B-tree library for OCaml Tom Ridge Wodan: a pure OCaml, flash-aware filesystem library Gabriel de Perthuis Tezos: the OCaml Crypto-Ledger Benjamin Canou et al.
14:40 15:30 - 15:30 15:50 16:10	 Stephen Dolan and KC Sivaramakrishnan Bioinformatics, The Typed Tagless Final Way Sebastien Mondet 16:30 Talk session 4 A B-tree library for OCaml Tom Ridge Wodan: a pure OCaml, flash-aware filesystem library Gabriel de Perthuis Tezos: the OCaml Crypto-Ledger Benjamin Canou et al. 17:40 Talk session 5
14:40 15:30 - 15:30 15:50 16:10 17:00 - 17:00	 Stephen Dolan and KC Sivaramakrishnan Bioinformatics, The Typed Tagless Final Way Sebastien Mondet 16:30 Talk session 4 A B-tree library for OCaml Tom Ridge Wodan: a pure OCaml, flash-aware filesystem library Gabriel de Perthuis Tezos: the OCaml Crypto-Ledger Benjamin Canou et al. 17:40 Talk session 5 Component-based Program Synthesis in OCaml
14:40 $15:30 - 15:30$ $15:50$ $16:10$ $17:00 - 17:00$ $17:90$	 Stephen Dolan and KC Sivaramakrishnan Bioinformatics, The Typed Tagless Final Way Sebastien Mondet 16:30 Talk session 4 A B-tree library for OCaml Tom Ridge Wodan: a pure OCaml, flash-aware filesystem library Gabriel de Perthuis Tezos: the OCaml Crypto-Ledger Benjamin Canou et al. 17:40 Talk session 5 Component-based Program Synthesis in OCaml Zhanpeng Liang and Kanae Tsushima Testing with Crowhar

Stephen Dolan and Mindy Preston

Erlang Workshop

09:00 – 09:10 **Opening & Welcome**

09:00	Opening & Welcome
-------	-------------------

09:10 – 10:00 Keynote Invited Talk

09:10 Keynote Martin Sumner

10:30 – 11:20 Session 2

- 10:30 Construction and Formal Verification of a Fault-Tolerant Distributed Mutual Exclusion Algorithm *Evgeniy Shishikin*
- 10:55 Towards an Isabelle/HOL Formalisation of Core Erlang Joseph Harrison

12:00 – 12:25 **Session 3**

12:00 In medias res: WIP discussion

14:00 - 14:50 **Session 4**

14:00 Distributed Memory Architecture for High-Level Synthesis of Embedded Controllers from Erlang

Kagumi Azuma, Nagisa Ishiura, Nobuaki Yoshida and Hiroyuki Kanbara Structuring Erlang BEAM control flow

14:25 Structuring Erlang BEAM control fl Dániel Lukács and Melinda Toth

15:30 – 16:20 **Session 5**

15:30	The Shared-Memory Interferences of Erlang/OTP Built-ins
	Stavros Aronis and Kostis Sagonas
15:55	Towards Change-driven Testing

Viktória Fördős, István Bozó and Melinda Toth

16:50 – 17:50 Session 6

- 16:50 eAOP An Aspect Oriented Programming Framework for Erlang Ian Cassar, Adrian Francalanza, Luca Aceto and Anna Ingolfsdottir
- 17:20 Erlang and Elixir development news (not yet confirmed)

CUFP Tutorials – Day 2

09:10 - 12:30	CUFP Tutorials C5	L5
09:10-10:00 10:30-11:30 12:00-12:30	Tutorial C5: Teaching Functional Programming Michael Sperber	
09:10 - 12:20	CUFP Tutorials C6	L4
09:10–10:00 10:30–11:30 12:00–12:30	Tutorial C6: Transducers in Practice Renzo Borgatti	
14:00 - 17:30	CUFP Tutorials C7	L5
14:00–15:00 15:30–16:20 16:50–17:30	Tutorial C7: Owl - Data Science in OCaml Liang Wang	
14:00 - 15:00	CUFP Tutorials C8	L4
14:00–15:00 15:30–16:20 16:50–17:30	Tutorial C8: GraphQL Servers in OCaml Andreas Garnæs	

Commercial Users of Functional Programming

09:10 - 10:00 CUFP Talks 1

09:10 Keynote

Bodil Stokke

09:35 Bonsai: a DSL for serverless firm real-time decisioning Jeremie Lasalle-Ratelle

10:30 – 11:20 **CUFP Talks 2**

- 10:30 Interfacing OCaml and Rust: picking the right tool for the job Joris Giovannangeli
- 10:55 Distributed load testing with MZBench Renat Idrisov

12:00 – 12:25 **CUFP Talks 3**

12:00 Gens N' Roses: Appetite for Reduction Jacob Stanley

14:00 – 14:50 **CUFP Talks 4**

- 14:00 Formally Verifying a Smart-Contract Language Implementation with Isabelle Simon Meier
- 14:25 Haskell games and apps for iOS and Android *Ivan Perez*

15:30 – 16:20 **CUFP Talks 5**

- 15:30 Using Haskell to run a datacenter Pavlo Kerestey
- 15:55 Functional Facades over Legacy Code Nicholas Cowle and Robin Kay

16:50 – 17:40 **CUFP Talks 6**

- 16:50 Building the largest payment sandbox on a tiny machine *Máté Marjai*
- 17:15 Using Functional Programming to Accelerate Translational Research at Pfizer Austin Huang

Haskell Implementors Workshop

09:10 - 10:10 State of GHC

09:10	Progress on GHC
00.20	Simon Peyton Jones
09.30	Ben Gamari
10:00	Getting Ready for Hadrian
	Andrey Mokhov, Zhen Zhang, Ben Gamari and Neil Mitchell
10:30 -	11:30 Compiling to LLVM
10:30	Native Support for Explicit Stacks in LLVM
	Kavon Farvardin and Simon Peyton Jones
10:55	SimplexHC: Lowering High-Level Haskell to Imperative IR
11.00	Siddharth Bhat
11:20	Lightning Talk Slot #1
12:00 -	12:25 Constraints
12:00	On Unsatisfiability
	J. Garrett Morris
14.00	15:00 Working in Core
14:00 -	15.00 working in Core
14:00	Why GHC Core and Linear Logic Should be Best Friends
14.05	Carter Schonwald and Joel Burget
14:25	Sebastian Craf
14:50	Lightning Talk Slot #2
15:30 -	16:30 Tool Support
15:30	IDE Support in GHC
	Alan Zimmannan
	zhan Zimmerman
15:55	Tracking GHC Performance
15:55	Tracking GHC Performance Mathieu Boespflug and Manuel Chakravarty
15:55 16:20	Tracking GHC Performance Mathieu Boespflug and Manuel Chakravarty Lightning Talk Slot #3

16:50 – 17:50 All Broken Up

- 16:50 An Experiment in Fragment-Based Code Distribution *Philipp Schuster*
- 17:15 Lightning Talk Slot #4

Functional Art, Music, Modeling and Design

09:00 - 09:10 Introduction

09:00 Welcome Michael Sperber and Jean Bresson

09:10 – 10:00 **Papers/Demos (Music)**

- 09:10 A Categorial Grammar for Music and Its Use in Automatic Melody Generation Halley Young
- 09:40 Demo | Representation of Musical Notation in Haskell Edward Lilley

10:30 – 11:30 **Demos (Music)**

- 10:30 Demo | The Arpeggigon: A Functional Reactive Musical Automaton Henrik Nilsson
- 10:50 Demo | Vivid: Sound Synthesis with Haskell and SuperCollider *Tom Murphy*
- 11:10 Demo | African Polyphony and Polyrhythm Chris Ford

12:00 – 12:30 **Paper**

12:00 Modelling the Way Mathematics Is Actually Done Joe Corneli, Ursula Martin, Dave Murray-Rust, Alison Pease, Raymond Puzio and Gabriela Rino Nesin

14:00 - 15:00 **Paper/Tutorial**

14:00 FAUST Tutorial for Functional Programmers Yann Orlarey, Stéphane Letz, Dominique Fober and Romain Michon

15:30 – 16:20 **Papers/Demos**

- 15:30 GALE: A Functional Graphic Adventure Library and Engine Ivan Perez
- 16:00 Demo | Ait: A Concatenative Language for Creative Programming Stian Veum Møllersen

16:50 – 17:40 **Papers/Demos**

- 16:50 Unified Media Programming: An Algebraic Approach Simon Archipoff and David Janin
- 17:20 Demo | Octopus: A High-Level Fast 3D Animation Language Simon Archipoff and David Janin

Takayuki Muranushi (1983–2017)

It is with great sadness that we have to report of the untimely passing of the CUFP Tutorial Co-Chair Takayuki Muranushi. We have lost an extraordinarily promising researcher and a true champion of functional programming.

Takayuki Muranushi was not trained as functional programmer: born in 1983, he entered Kyoto University in the physics and cosmology program. His PhD thesis (defended in 2013 at Kyoto University's Institute for Theoretical Physics) was titled "Lightning in Protoplanetary Disks". Writing code – in Fortran or, nowadays, C++, is part of contemporary physics practice. That was not enough for Takayuki. He went one level up: he got the computer to write C++ physics code, that was faster than any code a human could write, or comprehend.

I caught a glimpse of this metaprogram, Paraiso, from Takayuki's presentation at the 2012 Shonan meeting on code generation for high-performance computing, where I first met him. It is a Haskell program that takes a partial differential equation in familiar mathematical notation and, relying on sophisticated genetic algorithms, generates C/C++ code for a GPGPU or a multicore CPU. His aspiration, close to fulfillment, was to make Paraiso the standard for partial differential equations, as FFTW is for fast Fourier transforms. I was very impressed by the sophistication of Paraiso – and also by a rarely seen enthusiastic presentation of it. He had managed to fit 125 slides within allotted 10 minutes – in a way that I still remember them five years later.

Takayuki continued this work after graduation, at RIKEN Advanced Institute for Computational Science in the Particle Simulator Research Team. Last year he with his teammates presented a paper at the ICFP-affiliated FHPC workshop, on automatic generation of efficient code from mathematical descriptions of stencil computation. He has also co-authored several papers at the Haskell Symposium, and this year served on its Program Committee.

Takayuki not only studied functional programming: he lived it. In 2006, right after finishing his undergraduate studies, he co-founded a start-up 'Preferred Infrastructure (PFI)'. When I visited that company once, everyone I met was eager to tell me that although 'Preferred Infrastructure' is the official company name, what PFI really stands for is 'Pure Functional Inc.' – the name given by Takayuki. He is most well-known in Japan for the Japanese edition of Miran Lipovaca's book 'Learn You a Haskell', which he co-translated with his PFI colleague Hideyuki Tanaka.

He was so brilliant that he seemed not of this world. And now he isn't. His papers, his code, his automatic (robotic) book scanner, his book continue to surprise and delight. Those of us who were fortunate to have met him shall remember, and sorely miss him.

– Oleg Kiselyov

Notes

Location Map for Social Events



Platinum



Gold



Silver





Bronze





OBSIDIAN SYSTEMS

Microsoft Research Sell-Typed

The Haskell Consultants

ndustrial partners

Cover linocut by Jan Ritchie