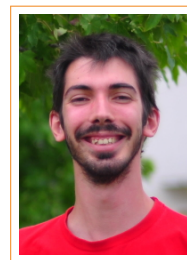


Gabriel Scherer

Parsifal, INRIA Saclay, France

✉ gabriel.scherer@inria.fr

🌐 www.gasche.info



(last update: May 10, 2024)

Position

- 2024–current Permanent researcher (CR [INRIA](#)), team Picube in [IRIF](#), Paris
- 2017–2023 Permanent researcher (CR [INRIA](#)), team Partout in Saclay
- 2016–2017 Post-doc working with [Amal Ahmed](#) in the [PRL group](#) at Northeastern University, Boston
- 2012–2016 PhD student under the supervision of [Didier Rémy](#), team [Gallium](#) in Rocquencourt.

Research areas

I work in programming language research, with two main areas:

1. type theory and proof theory, in particular applying focusing to programming-language questions such as program equivalence;
2. the evolution of the OCaml programming language, in particular its type system.

Software

I am a co-maintainer of the [OCaml compiler distribution](#), the main implementation of the OCaml programming language. A significant portion of my working time is spent proposing improvements to the implementation or reviewing and integrating changes proposed by other people.

For more details, see [my Software webpage](#).

Teaching

- 2023–2024 Teacher: logical relations (master, M2, 10h only)
- 2021–2022 Teacher: effects in programming languages (master, M2, 10h only)
- 2021–2022 Teacher: intro to functional programming: OCaml (bachelor, L1)
- 2020–2021 Teacher: intro to functional programming: OCaml (bachelor, L1)
- 2019–2020 Teacher: intro to functional programming: OCaml (bachelor, L1)
- 2018–2019 Teacher: intro to functional programming: Racket (bachelor, L1)
- 2014–2015 Teaching assistant: Java (bachelor, L1), C (bachelor. L3)
- 2013–2014 Teaching assistant: Java (bachelor, L1), Advanced Functional Programming (master, M1)
- 2008–2010, 2012 Teaching assistant: Caml-Light (bachelor, L1)
- 2006–now Helping beginners learn programming in all kinds of online forums

Community service

Scientific events

- 2023 Chair for the OCaml Workshop 2023
- 2023 Program Committee member for ML Workshop 2023
- 2022 Program Committee member for the TyDe'22 and LFMTTP'22 workshops

- 2022 Program Committee member for the ICFP'22 conference
- 2022 Co-chair for the Artifact Evaluation Committee of ICFP'22
- 2021 Program Committee member for TYPES'21
- 2021 Co-chair for the Artifact Evaluation Committee of ICFP'21
- 2020 Co-chair for the PLMW workshop at POPL'21
- 2018 Program Committee member for the POPL'19 conference
- 2018 Chair for the ML Family Workshop 2018
- 2017 Program Committee member for the ICFP'17 conference
- 2017 Chair for the OCaml Workshop 2017

Other

- 2019-today Director of the [OCaml Software Foundation](#). This Foundation collects around 200K€ each year from industrial sponsors, and we are in charge of providing this funding to projects improving the OCaml ecosystem and community.
- 2017-2023 Moderator of the “types” and “types-announce” mailing-lists (4165 members in 2023). In 2021 I created a corresponding Zulip Channel, <https://typ.zulipchat.com/> (480 users).

International publications

- 2024 “[Unboxed Data Constructors: Or, How cpp Decides a Halting Problem](#)”. By Nicolas Chataing, Stephen Dolan, Gabriel Scherer, and Jeremy Yallop. *POPL*.
- 2023 “[Coqlex: Generating Formally Verified Lexers](#)”. By Wendlasida Ouedraogo, Lutz Strassburger, and Gabriel Scherer. *Programming*.
- 2022 “[Debootstrapping without Archeology - Stacked Implementations in Camlboot](#)”. By Nathanaëlle Courant, Julien Lepiller, and Gabriel Scherer. *Programming*.
- 2021 “[A practical mode system for recursive definitions](#)”. By Gabriel Scherer Alban Reynaud and Jeremy Yallop. *POPL*.
- 2019 “[Functional programming with lambda-tree syntax](#)”. By Dale Miller Ulysse Gérard and Gabriel Scherer. *PPDP*.
- 2018 “[Capturing the Future by Replaying the Past – Functional Pearl](#)”. By James Koppel, Gabriel Scherer, and Armando Solar-Lezama. *ICFP*.
“[Correctness of Speculative Optimizations with Dynamic Deoptimization](#)”. By Olivier Flückiger, Gabriel Scherer, Ming-Ho Yee, Aviral Goel, Amal Ahmed, and Jan Vitek. *POPL*.
“[FabULous Interoperability for ML and a Linear Language](#)”. By Gabriel Scherer, Max S. New, Nicholas Rioux, and Amal Ahmed. *FoSSaCS*.
“[Merlin: a Language Server for OCaml \(Experience report\)](#)”. By Frédéric Bour, Thomas Refis, and Gabriel Scherer. *ICFP*.
- 2017 “[Deciding equivalence with sums and the empty type](#)”. By Gabriel Scherer. *POPL*.
- 2015 “[Full reduction in the face of absurdity](#)”. By Gabriel Scherer and Didier Rémy. *ESOP*.
“[Multi-focusing on extensional rewriting with sums](#)”. By Gabriel Scherer. *TLCA*.
“[Polarised Intermediate Representation of Lambda Calculus with Sums](#)”. By Guillaume Munch-Maccagnoni and Gabriel Scherer. *LICS*.
“[Which simple types have a unique inhabitant?](#)” By Gabriel Scherer and Didier Rémy. *ICFP*.
- 2013 “[GADTs meet subtyping](#)”. By Gabriel Scherer and Didier Rémy. *ESOP*.
“[Tracking Data-Flow with Open Closure Types](#)”. By Gabriel Scherer and Jan Hoffmann. *LPAR*.
- 2012 “[On Irrelevance and Algorithmic Equality in Predicative Type Theory](#)”. By Andreas Abel and Gabriel Scherer. *Logical Methods in Computer Science* (2012).

National publications

- 2024 “[Correct tout seul, sûr à plusieurs](#)”. By Clément Allain and Gabriel Scherer. *JFLA*.
- 2023 “[Backtracking reference store](#)”. By Gabriel Scherer Camille Noüs. *JFLA*.
- 2022 “[Déboîter les constructeurs](#)”. By Gabriel Scherer Nicolas Chataing Camille Noüs. *JFLA*.
- 2021 “[Tail Recursion Modulo Cons](#)”. By Gabriel Scherer Frédéric Bour Basile Clément. *JFLA*.
- 2019 “[Unboxing Mutually Recursive Type Definitions in OCaml](#)”. By Rodolphe Lepigre Simon Colin and Gabriel Scherer. *JFLA*.
- 2017 “[Search for Program Structure](#)”. By Gabriel Scherer. *SNAPL*.
- 2015 “[Normalization by realizability also evaluates](#)”. By Pierre-Évariste Dagand and Gabriel Scherer. *JFLA*.
- 2010 “[Macaque: Interrogation sûre et flexible de bases de données depuis OCaml](#)”. By Gabriel Scherer and Jérôme Vouillon. *JFLA*.

Presentations in peer-reviewed workshops

- 2023 “The combinatorics of free bifibrations”, Bryce Clarke, Gabriel Scherer, Noam Zeilberger, CLA 2023
- 2022 “Boxroot, fast movable GC roots for a better FFI”, Guillaume Munch-Maccagnoni, Gabriel Scherer, ML Workshop 2022
- 2022 “An OCaml use case for strong call-by-need reduction”, Gabriel Scherer, Nathanaëlle Courant, ML Workshop 2022
- 2021 “Unfolding ML datatype declarations without loops”, Nicolas Chataing, Gabriel Scherer, ML Workshop 2021
- 2021 “Frozen inference constraints for type-directed disambiguation”, Olivier Martinot, Gabriel Scherer, ML Workshop 2021
- 2020 “Quantified applicatives: API design for type-inference constraints”, Olivier Martinot, Gabriel Scherer. ML Workshop 2020
- 2020 “Translation validation of a pattern-matching compiler”, Francesco Mecca, ML Workshop 2020
- 2016 “[Ambiguous pattern variables](#)”, Gabriel Scherer, Luc Maranget and Thomas Réfis, ML Workshop 2016
- 2014 “Deciding unique inhabitants with sums”, Gabriel Scherer, TYPES 2014
“[Github Pull Requests for OCaml development: a field report](#)”, Gabriel Scherer, OCaml Workshop 2014
“[Well-typed generic smart-fuzzing for APIs](#)”, Thomas Braibant (presenter), Jonathan Protzenko and Gabriel Scherer, ML Workshop 2014
- 2013 “Mining opportunities for unique inhabitants in dependent programs”, Gabriel Scherer, Dependently Typed Programming Workshop (DTP) 2013
- 2012 “GADT meet subtyping”, Gabriel Scherer and Didier Rémy, ML Workshop 2012

Thesis

[Which types have a unique inhabitant?](#)
[Focusing on pure program equivalence](#)

Supervised by [Didier Rémy](#).
Defended on March 30th, 2016.