Meven Lennon-Bertrand

Post-doctoral researcher – University of Cambridge

mgapb2@cam.ac.uk www.meven.ac MevenBertrand

Academic Positions

Research Associate

Principal Investigator: Neel Krishnaswami

Nov. 2022—... University of Cambridge

Under the research project *TypeFoundry – Foundations of Type Inference for Modern Programming Languages*.

PhD Student

Supervised by Nicolas Tabareau

Sep. 2019—Sep. 2022 Gallinette team, Nantes

Research Intern

Supervised by Nicolas Tabareau

Feb.—Jul. 2019 Gallinette team, Nantes

Gradualizing the Calculus of Inductive Constructions.

Research Intern

Supervised by Jurriaan Rot

Jan.—Jul. 2017 Radboud University Nijmegen

Coalgebraic Determinization of Alternating Automata.

Main Research Contributions

Thesis

Lennon-Bertrand, *Bidirectional Typing for the Calculus of Inductive Constructions* (PhD Thesis, 2022)

Journal Articles

Lennon-Bertrand et al., *Gradualizing the Calculus of Inductive Constructions* (TOPLAS, 2022)

Conference Articles

- Lennon-Bertrand, *Complete Bidirectional Typing for the Calculus of Inductive Constructions* (ITP, 2021)
- Maillard et al., *A Reasonably Gradual Type Theory* (ICFP, 2022)

Formalisations

- MetaCoq (contributor): formalisation of Coq's meta-theory and of a certified type-checker.
- LogRel-Coq (leader): formalisation of logical relation proofs for dependent type systems.

Interests

Proof Assistants | Logic | Dependent Types | Bidirectional Typing | Formalisation of Mathematics

Education

Master 2 (Computer Science)

2018—2019 ENS de Lyon

Master 2 (Mathematics)

Preparation to the Agrégation, received 10th

2017—2018 ENS de Lyon

Master 1 (Mathematical Foundations of Computer Science)

Erasmus exchange

2016—2017 Nijmegen, NL

Bachelor (Computer Science & Mathematics)

Double Bachelor

2015—2016 ENS de Lyon

Teaching and Outreach

Lecturer: Denotational Semantics

Fall 2023 University of Cambridge

Teaching Assistant

Sep. 2019—Jun. 2022 Université de Nantes

CHantiers Arts, Sciences et Technologies

2019–2022 Lycée Michelet, Nantes

Collaborated with a maths researcher and two theatre authors on a project with a vocational high-school.

Past Research Results

Topic

My research primarily aims at taking concepts and ideas originating in programming language (PL) theory, and applying them to proof assistants based on type theory, and reciprocally of turning practical proof assistant questions into type theoretic or PL ones, in order to better understand and solve them. Difficult technical questions typically arise when adapting concepts the simple type systems of the standard PL setting to the very rich and expressive ones on which proof assistants rely. Most of my past work has been focused on solving this kind of questions. Although I mainly work in and on Coq and its type system, the Calculus of Inductive Constructions, most ideas apply directly to other type-theoretic proof assistants, such as Lean or Agda.

Main Theoretical Results

Gradualization of the Calculus of Inductive Constructions

[Len+22; Mai+22] Kenji Maillard, Nicolas Tabareau, Éric Tanter

We proposed a system extending the Calculus of Inductive Constructions to feature a form of dynamic typing, following ideas from the gradual typing research area. We first proved an impossibility theorem, showing that one cannot to design a system combining all the desirable properties for such a system. Still, we were able to design a solution where one can modularly choose the property to be abandoned and those to keep. Later on, we extended that system to allow for sound, internal reasoning about gradual programs.

The practical goal is to provide support for smooth development, in particular by giving better support to working with uncompleted proofs and terms. While I am not active on this topic any more, work is still ongoing in turning our theoretic work into an extension for Coq.

Bidirectional Calculus of Inductive Constructions

[Len21; SLF22]

Bidirectional typing is an approach which makes explicit in the typing rules the flow of information present in implementations. While it belongs to the folklore of dependent type systems implementors, there was surprisingly little theoretical work on it. I gave a precise bidirectional presentation for the Calculus of Inductive Constructions, and proved its properties, including its equivalence with the standard presentation. I formalised this equivalence in the MetaCoq project, and built on it to show the completeness of the certified type-checker which is at the heart of the project.

The formalisation led to the discovery of a bug in the type-checker of Coq – see Issue #13495 and Sozeau et al. [SLF22] –, demonstrating the value of the certification work undertaken in MetaCoq.

Software Development: Formalisation Projects

MetaCoq

2020-... The MetaCoq team

This large collaborative project aims at formalizing Coq in Coq itself, and to allow manipulating Coq terms in Coq in order to develop certified meta-programming tools. I mainly contribute to the theoretical aspect, and currently my largest addition is the formalisation of my work on bidirectional typing, in order to prove the correctness of the type-checking algorithm implemented as part of the project.

This work allowed detecting and fixing a bug in the kernel of Coq. MetaCoq's certified type-checker serves as a basis for the certified compilers CertiCoq – from Coq to CompCert's Clight – and ConCert – to Elm, Rust and smart contract languages.

LogRel-Coq

2022-... A. Adjedj, K. Maillard, P.-M. Pédrot, L. Pujet

This project aims at formalising, in Coq, proofs by logical relations of deep meta-theoretic

properties of dependent type systems, in particular normalisation. It is complementary to MetaCoq, in the sense that the latter concentrates on obtaining a certified type-checker for a system that is as close as possible to "real" Coq, but admits normalisation. Instead, LogRel-Coq currently works on a more toyish language, but avoids any form of axiom.

Research Activity

Supervised students

Robin Jourde – Understanding the η Law for Functions in CIC

Master 2 internship

Jan. – Jul. 2023 | Co-supervised with Nicolas Tabareau

Matthew Sirman – A Normalisation by Evaluation Implementation of a Type Theory with Observational Equality

Part III dissertation (4th, final, undergrad year)

Nov. 2022 – May 2023 | Co-supervised with Neel Krishnaswami

Matthew was distinguished as the best overall student in Part III Computer Science, in part for his dissertation.

Research Visits & Invited Seminars

Meta-programming, quoting, and modalities in type theory – University of Nantes

16 – 20 Oct. 2023 | Invited by Yannick Forster

Week-long mini-workshop. Also gave a seminar talk.

Mathematically Structured Programming Group – University of Strathclyde

26 – 30 Jun. 2023 | Invited by Conor McBride

CHoCoLa Seminar

Jan. 2023 | ENS de Lyon

CHoCoLa is a monthly seminar organized by the Plume team of the ENS de Lyon, federating the French community around logic and computation.

LoVE Team Seminar

Dec. 2022 | Université Sorbonne Paris Nord

Faculty of Mathematics and Physics – University of Ljubljana

9 – 13 May 2022 | Invited by Andrej Bauer

Academic and Community Service

Normalisation by Evaluation Reading Group

2023-...

Organized a reading group in the CLASH team on normalisation by evaluation.

Artefact Evaluation for ICFP

2022

Evaluation of artefacts (code, formalisation, supplementary material, etc.) attached to articles accepted at the conference.

Proof Assistants Stack Exchange

2022-...

This website aims to answer questions around proof assistants in a community-based manner. I am in the top 5 most active users.

Elected Student Representative

2017-2018 ENS de Lyon

Publications

Most of my publications list authors in alphabetical order. Exceptions are the ICFP 2019 article, where Kenji Maillard was the main investigator, and the 2023 preprint where Matthieu Sozeau, as the leader of the MetaCoq project, appears as first author.

Journal

[Len+22] Gradualizing the Calculus of Inductive Constructions
2022 Meven Lennon-Bertrand, Kenji Maillard, Nicolas Tabareau and Éric Tanter. *ACM Transactions on Programming Languages and Systems*, 82 pp.

Conference

[Mai+22] A Reasonably Gradual Type Theory
2022 Kenji Maillard, Meven Lennon-Bertrand, Nicolas Tabareau and Éric Tanter. *International Conference on Functional Programming*, 29 pp.

[Len21] Complete Bidirectional Typing for the Calculus of Inductive Constructions
2021 Meven Lennon-Bertrand. *12th International Conference on Interactive Theorem Proving*, 19 pp.

Peer-reviewed workshops

[LK23] Decidable Type-Checking for Bidirectional Martin-Löf Type Theory
2023 Meven Lennon-Bertrand and Neel Krishnaswami. *29th International Conference on Types for Proofs and Programs*.

[Adj+23] Engineering logical relations for MLTT in Coq
2023 Arthur Adjedj, Meven Lennon-Bertrand, Kenji Maillard and Loïc Pujet. *29th International Conference on Types for Proofs and Programs*.

[Len22a] Equivalence between Typed and Untyped Algorithmic Conversions
2022 Meven Lennon-Bertrand. *28th International Conference on Types for Proofs and Programs*.

[Len22b] À bas l'η – Coq's troublesome η-conversion
2022 Meven Lennon-Bertrand. *1st Workshop on the Implementation of Type Systems*.

[SLF22] The Curious Case of Case: Correct & Efficient Representation of Case Analysis in Coq and MetaCoq
2022 Matthieu Sozeau, Meven Lennon-Bertrand and Yannick Forster. *1st Workshop on the Implementation of Type Systems*.

Preprints under review

[Adj+23] Martin-Löf *à la* Coq
2023 Arthur Adjedj, Meven Lennon-Bertrand, Kenji Maillard, Pierre-Marie Pédrot and Loïc Pujet.

[LLM23] Definitional Functoriality for Dependent (Sub)Types
2023 Théo Laurent, Meven Lennon-Bertrand and Kenji Maillard.

[Soz+23] Correct and Complete Type Checking and Certified Erasure for Coq, in Coq
2023 Matthieu Sozeau, Yannick Forster, Meven Lennon-Bertrand, Jakob Botch Nielsen, Nicolas Tabareau and Théo Winterhalter.

Unpublished

[BR18] Coalgebraic Determinization of Alternating Automata
2018 Meven Bertrand and Jurriaan Rot. *arXiv*.

Teaching and Science Outreach

Lecturer: Denotational Semantics

Fall 2023 University of Cambridge

I taught the Denotational Semantics course to Part II (3rd year) University of Cambridge students. The course content was based on previous iterations by Marcelo Fiore, Andy Pitts and Glynn Winskel.

Teaching Assistant (64 hrs/year)

Sep. 2019—Jun. 2022 Université de Nantes

During my 3 years as a Ph.D. student, I also served as a teaching assistant at the University of Nantes. I taught various levels (1st to 3rd Bachelor years), themes (mathematics, applied and fundamental computer science), formats (lectures, exercise and computer sessions), and publics (specialists and non-specialists).

CHantiers Arts, Sciences et Technologies

2019–2022 Théâtre Athénor & Lycée Professionnel Michelet, Nantes

Together with math researcher Bertrand Michel and theatre authors Rémi Checchetto and Sylvain Renard, we collaborated with a vocational high-school teachers to build workshops for their students around the theme of "Artificial Intelligence", in a broad sense. I implemented activities directly inspired by the *Computer Science Unplugged* project, and designed some of my own. These workshops culminated in an exhibition, created by the students, based on the content of the workshops.

Séminaire de la Détente Mathématique

2018—2019 Maison des Mathématiques et de l'Informatique, Lyon

A weekly mathematic/computer science seminar, aimed at being "relaxed" and accessible to both students and faculty, with talks often on unusual and/or fun topics. Many students would also give their first actual talk there. I organized the seminar with a team of other students, and participated as an orator.

Category Theory Course

Sept 2018 – Jan 2019 ENS de Lyon

During my Master 2, I co-organized a category theory course for students of the ENS de Lyon. Because of the absence from the curriculum of a subject that was nonetheless required in some master courses, a fellow student and I decided to set it up a course on the subject, with which we were familiar thanks to previous internships. Although it was not integrated to the official curriculum, we kept it as close as possible to a proper lecture: it lasted for a semester of a 2-hour lecture per week (split between us two), with a few dozen students, and we covered a large part of Awodey's *Category theory*.