

# Lionel Parreaux | Curriculum vitae

Route de Chavannes 68 – 1007 Lausanne – Switzerland

☎ +41 79 135 48 94 • ✉ [lionel.parreaux@gmail.com](mailto:lionel.parreaux@gmail.com)  
🌐 [lptk.github.io/about](https://lptk.github.io/about) • [in lparreaux](#) • [🔗 LPTK](#)

## Education

---

<b>EPFL (Swiss Federal Institute of Technology)</b> <i>Ph.D. in Computer Science</i>	<b>Lausanne</b> 2014–2020
<b>NUS (National University of Singapore)</b> <i>Academic exchange (1 semester)</i>	<b>Singapore</b> Fall 2013
<b>INSA Lyon (National Institute of Applied Science)</b> <i>M.Sc. (Engineering Degree) in Computer Science</i>	<b>Lyon</b> 2009–2014

**Languages:** French (mother tongue) English (bilingual) Spanish (intermediate)

**Research interests:** Programming languages, type systems, compiler design, domain-specific languages, and database technology. I believe that improving the performance, safety, and usability of high-level programming is essential to the future of software engineering as a whole.

## Publications

---

- **Lionel Parreaux** and Amir Shaikhha. 2020. **Multi-stage Programming in the Large with Staged Classes**. In Proceedings of the 19th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences (**GPCE 2020**). DOI: <https://doi.org/10.1145/3425898.3426961>
- **Lionel Parreaux**. 2020. **The Simple Essence of Algebraic Subtyping: Principal Type Inference with Subtyping Made Easy (Functional Pearl)**. In Proc. ACM Program. Lang. 4, ICFP, Article 124 (**ICFP 2020**). DOI: <https://doi.org/10.1145/3409006>
- Amir Shaikhha and **Lionel Parreaux**. 2019. **Finally, a Polymorphic Linear Algebra Language**. In 33rd European Conference on Object-Oriented Programming (**ECOOP 2019**). DOI: <https://doi.org/10.4230/LIPIcs.ECOOP.2019.25>
- **Lionel Parreaux**, Aleksander Boruch-Gruszecki, and Paolo G. Giarrusso. 2019. **Towards improved GADT reasoning in Scala**. In Proceedings of the Tenth ACM SIGPLAN Symposium on Scala (**SCALA 2019**). DOI: <https://doi.org/10.1145/3337932.3338813>
- **Lionel Parreaux** and Christoph E. Koch. 2018. **Comprehending Monoids with Class** (Extended Abstract). In Proceedings of Type-Driven Development (**TyDe 2018**). <https://icfp18.sigplan.org/details/tyde-2018/12>
- **Lionel Parreaux**, Antoine Voizard, Amir Shaikhha, and Christoph E. Koch. 2018. **Unifying Analytic and Statically-Typed Quasiquotes**. In Proc. ACM Program. Lang. (**POPL 2018**). DOI: <https://doi.org/10.1145/3158101>
- **Lionel Parreaux**, Amir Shaikhha, and Christoph E. Koch. 2017. **Quoted staged rewriting: a practical approach to library-defined optimizations**. In Proceedings of the 16th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences (**GPCE 2017**). DOI: <https://doi.org/10.1145/3136040.3136043>
- **Lionel Parreaux**, Amir Shaikhha, and Christoph E. Koch. 2017. **Squid: type-safe, hygienic, and reusable quasiquotes**. In Proceedings of the 8th ACM SIGPLAN International Symposium on Scala (**SCALA 2017**). DOI: <https://doi.org/10.1145/3136000.3136005>

- Amir Shaikhha, Yannis Klonatos, **Lionel Parreaux**, Lewis Brown, Mohammad Dashti, and Christoph Koch. 2016. **How to Architect a Query Compiler**. In Proceedings of the 2016 International Conference on Management of Data (**SIGMOD 2016**). DOI: <https://doi.org/10.1145/2882903.2915244>

## Recognition

---

### Awards.....

- (2017) **GPCE Best Paper Award** (*Quoted Staged Rewriting* paper).
- (2014) EPFL **EDIC PhD program fellowship**.

### Presentations, Seminars, and Invitations.....

- (July 2019) *Towards improved GADT reasoning in Scala*. Conference talk, SCALA.
- (June 2018) *Fearless Metaprogramming with Squid*. Invited talk, DIMA lab, TU Berlin.
- (June 2018) *Fearless Metaprogramming with Squid*. Invited talk, Amazon Berlin.
- (September 2018) *Comprehending Monoids with Class*. Type-Driven Development, St. Louis.
- (January 2018) *Unifying analytic and statically-typed quasiquotes*. Conference talk, POPL.
- (December 2017) *Unifying analytic and statically-typed quasiquotes*. Invited talk, EPFL LAMP.
- (October 2017) *Quoted Staged Rewriting: a Practical Approach to Library-Defined Optimizations*. Conference talk, GPCE.
- (October 2017) *Squid: Type-Safe, Hygienic, and Reusable Quasiquotes*. Conference talk, SCALA.
- (September 2017) *Quoted Staged Rewriting: a Practical Approach to Library-Defined Optimizations*. Invited talk, EPFL LAMP.
- (2017, 2018, 2019) Google Compiler and Programming Language Summit, Munich.
- (2016) Google PhD Student Summit on Compiler & Programming Technology, Munich.

### Open Source Contributions.....

- **Simple-sub** (52 stars)  <https://github.com/LPTK/simple-sub>
- **Squid** (171 stars)  <https://github.com/epfldata/squid>
- **dbStage** (12 stars)  <https://github.com/epfldata/dbstage>
- **Boilerless** (38 stars)  <https://github.com/lptk/boilerless>

### References.....

- **Christoph E. Koch**, EPFL, Lausanne. [christoph.koch@epfl.ch](mailto:christoph.koch@epfl.ch)  
<https://people.epfl.ch/christoph.koch>
- **Simon Peyton Jones**, Microsoft Research, Cambridge. [simonpj@microsoft.com](mailto:simonpj@microsoft.com)  
<https://www.microsoft.com/en-us/research/people/simonpj/>
- **Martin Odersky**, EPFL, Lausanne. [martin.odersky@epfl.ch](mailto:martin.odersky@epfl.ch)  
<https://lampwww.epfl.ch/~odersky/>
- **Viktor Kuncak**, EPFL, Lausanne. [viktor.kuncak@epfl.ch](mailto:viktor.kuncak@epfl.ch)  
<http://lara.epfl.ch/~kuncak/>

## Experience

---

### Research.....

#### Research Intern, Optimization and Spreadsheets

EPFL, Lausanne

Microsoft Research, Cambridge

Summer 2018 (3 months)

- Started the design and implementation of a novel intermediate representation (IR) for optimizing pure functional languages, based on a graph representation with incremental substitution constructs.
- Designed a domain-specific language (DSL) for dynamic programming, as a way to capture common patterns of spreadsheet formulae, and worked on using the graph IR to optimize that DSL.

#### PhD Semester Project, Metaprogramming Tools

EPFL, Lausanne

Data Analysis Theory and Applications Laboratory (DATA Lab)

Spring 2015

- Implemented a quasiquotation engine for SC (Systems/Compiler co-design framework written in *Scala*), making use of advanced macros and type introspection.
- Benchmarked the macro implementation to optimize hot spots and enhance the user experience.

Report [available on this link](#).

#### PhD Semester Project, Type Systems

EPFL, Lausanne

Lab for Automated Reasoning and Analysis (LARA Lab)

Fall 2014

- Formalized a novel Type and Effect System based on regular-expression regions.
- Proved its safety regarding memory management (no dangling pointers; no memory leaks).

Report [available on this link](#).

#### Research Intern, Type Systems

EPFL, Lausanne

Lab for Automated Reasoning and Analysis (LARA)

Summer 2014 (5 months)

- Explored the design and implementation of *Seagl*, a programming language I designed allowing safe memory management without garbage collection, thanks to an effect system. The report is [available on this link](#).

### Industry.....

#### R&D Intern, C++ Research Engineering

Palaiseau

Thales Research & Technology

Summer 2013 (3 months)

- Refactored and improved the usability of the open source *C++* library *paradiseo*.<sup>1</sup>
- Implemented/tested new simulated annealing algorithm from a research paper; it was added to *paradiseo*.

#### Developer Intern, Build Systems and Testing

Boulogne Billancourt

DxO Labs (Image processing software)

Summer 2012 (3 months)

- Set up a regression testing framework in *Python*, generating results in *HTML5*.
- Coded a build system using *CMake* and *Python*.
- Integrated these critical tools, significantly increasing the productivity of core developers (8 to 10 persons).

### Teaching Assistantship.....

CS-452: Foundations of Software (2019)

CS-449: Systems for Data Science (2018, 2019)

CS-210: Functional Programming (2018)

CS-251: Theory of Computation (2017)


CS-422: Database Systems (2016)

CS-110: Information, Computation, Communication (2016)

CS-111: Programming I (2015, 2017)

MATH-186: Mathematics II (2015)

---

<sup>1</sup> Paradiseo:  <https://github.com/nojhan/paradiseo>

## Notable projects.....

- (2017–Present) Implementing **dbStage**,<sup>2</sup> a **staged database compilation framework** based on Squid. The goal of dbStage is to allow programmers to embed low-footprint database systems right inside their applications, with no impedance mismatch, all the while benefitting from the usual advanced database optimization techniques.
- (2016–Present) Developed the **Squid**<sup>3</sup> **type-safe metaprogramming** framework for Scala, which extends the state of the art in multi-stage programming in several directions: it allows for **pattern matching and rewriting** existing code; it guarantees **type- and scope-safety** of metaprograms; it adds support for manipulating not only expressions but also definitions like classes and methods.
- (2018) As part of the **Microsoft Hackathon 2018**, created a language called MLScript which implemented **MLsub type inference**, compiled to Javascript, and could extract type information from TypeScript libraries for interoperability.
- (2016) Developed **Boilerless**,<sup>4</sup> a macro annotation that makes defining Scala class hierarchies more concise, and which influenced the design of the later enum syntax in Scala 3.
- (2014) **Led development** of the final subject of the *Cod'INSA 2014 programming contest* – a real-time multiplayer game interacting with *Java*, *Python* and *C++* artificial intelligences written by the candidates, using *Apache Thrift*, *Swing*, and a web-based interface.<sup>5</sup>  
The project was a success and allowed ranking the different teams according to their results.

## References.....

Available on page 2.

---

<sup>2</sup> dbStage: <https://github.com/epfldata/dbstage>

<sup>3</sup> Squid: <https://github.com/epfldata/squid>

<sup>4</sup> Boilerless: <https://github.com/LPTK/Boilerless>

<sup>5</sup> Cod'INSA final 2014: <https://github.com/cod-insa/cod-insa-2014>