Lionel Parreaux | Curriculum vitae

Room 3547, CSE, HKUST – Clear Water Bay – Hong Kong SAR
(852) 2358 8341 • parreaux@ust.hk
https://cse.hkust.edu.hk/~parreaux • lparreaux • LPTK

Employment

HKUST (The Hong Kong University of Science and Technology) — **Hong Kong**
Assistant Professor, *Department of Computer Science and Engineering (CSE)* — *February 2021–present*

Academic Qualifications

EPFL (Swiss Federal Institute of Technology) — **Lausanne**
Ph.D. in Computer Science — *2014–2020*

NUS (National University of Singapore) — **Singapore**
Academic exchange (1 semester) — *Fall 2013*

INSA Lyon (National Institute of Applied Science, Lyon) — **Lyon**
Master-level engineering degree — *2009–2014*

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

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

Summary of Achievements at HKUST

During my time at HKUST, I have been actively involved in research, teaching, and service.

Research:
- Established a research group, recruiting 10 PhD students (**5 with HKPFS**) and 4 MPhil students.
- Graduated 4 MPhil students and had 3 PhD students pass their PhD qualifying exams.
- Had **10 papers** accepted at the **top-ranking venues** of Programming Languages research, including 6 at OOPSLA, 2 at POPL, 1 at ICFP, and 1 at ECOOP. Also had 3 papers at well-known but less prestigious conferences (FLOPS and GPCE) and 3 at refereed workshops, for a **total of 16** publications.
- Received the **Distinguished Paper** award at ICFP 2024 and OOPSLA 2024 (x2), and the **Distinguished Artifact** award at ECOOP 2023 and OOPSLA 2025. Both distinctions are awarded to *at most* 10% of all *accepted* submissions (upper bound), which is often significantly fewer in practice.
- Secured a total of **HKD 2,387,167** research funding as sole PI, including an **RGC ECS** grant.
- Established ties with two companies, leading to two **joint research projects** and to technology transfer.
- Gave one **distinguished speaker talk** and **7 invited talks** at international academic and industrial venues, including the Universities of Cambridge, Edinburgh, and Hong Kong.

Teaching:
- Designed **three new courses**, including a course on compilers, which was **sorely missing** from our curriculum.
- Received **very positive** (well-above average) student feedback scores for all 6 taught semesters *but one*.
- Gave an **invited lecture** at the University of Edinburgh.

Service:
- Supervised numerous undergraduate (22) and graduate (11) students in their research projects.
- Was invited to the **program committee** of 10 conferences and workshops, including **two top conferences** (OOPSLA, POPL, and ICFP); was invited as an external reviewer at two conferences.

Peer-Reviewed Publications

Note: The *Proceedings of the ACM on Programming Languages* is a journal that publishes the proceedings of major Programming Languages conferences. Only the conference versions of relevant publications are listed here.
Typographic conventions: Graduate advisees are in *italics*. Papers published while at HKUST start with ⋆.

Refereed Publications in Top Venues

*The following are **A-ranked venues** in programming languages at the time of paper acceptance.*

- ⋆ *Chun Yin Chau* and **Lionel Parreaux**. 2026. The Simple Essence of Boolean-Algebraic Subtyping: Semantic Soundness for Algebraic Union, Intersection, Negation, and Equi-recursive Types. To appear in Proc. ACM Program. Lang. **POPL**.
- ⋆ *Cunyuan Gao* and **Lionel Parreaux**. 2025. A Lightweight Type-and-Effect System for Invalidation Safety. Proc. ACM Program. Lang. 8, **OOPSLA**2 (October 2025).
  🏆 **Distinguished Artifact**
- ⋆ *Luyu Cheng* and **Lionel Parreaux**. 2024. The Ultimate Conditional Syntax. Proc. ACM Program. Lang. 8, **OOPSLA**2 (October 2024).
  🏆 **Distinguished Paper**
- ⋆ † Amir Goharshady, *Chun Kit Lam*, **Lionel Parreaux**. 2024. Fast and Optimal Extraction for Sparse Equality Graphs. Proc. ACM Program. Lang. 8, **OOPSLA**2 (October 2024).
  († authors in alphabetical order)
  🏆 **Distinguished Paper**
- ⋆ *Yijia Chen* and **Lionel Parreaux**. 2024. The Long Way to Deforestation: A Type Inference and Elaboration Technique for Removing Intermediate Data Structures. Proc. ACM Program. Lang. 8, **ICFP**, Article 245 (August 2024), 35 pages. 10.1145/3674634
  🏆 **Distinguished Paper**
- ⋆ **Lionel Parreaux**, Aleksander Boruch-Gruszecki, *Andong Fan*, *Chun Yin Chau*. 2024. When Subtyping Constraints Liberate: A Novel Type Inference Approach for First-Class Polymorphism. Proc. ACM Program. Lang. 8, **POPL**, Article 48 (January 2024), 33 pages. 10.1145/3632890
- ⋆ *Ishan Bhanuka*, **Lionel Parreaux**, David Binder, and Jonathan Immanuel Brachthäuser. 2023. Getting into the Flow: Towards Better Type Error Messages for Constraint-Based Type Inference. Proc. ACM Program. Lang. 7, **OOPSLA**2, Article 237 (October 2023), 29 pages. 10.1145/3622812
- ⋆ *Andong Fan* and **Lionel Parreaux**. super-Charging Object-Oriented Programming Through Precise Typing of Open Recursion. In 37th European Conference on Object-Oriented Programming (**ECOOP 2023**). Leibniz International Proceedings in Informatics (LIPIcs), Volume 263, pp. 11:1-11:28, Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2023). 10.4230/LIPIcs.ECOOP.2023.11
  🏆 **Distinguished Artifact**
- ⋆ **Lionel Parreaux**, *Chun Yin Chau*. 2022. MLstruct: principal type inference in a Boolean algebra of structural types. Proc. ACM Program. Lang. 6, **OOPSLA**2, Article 141 (October 2022), 30 pages. 10.1145/3563304
- ⋆ Aleksander Boruch-Gruszecki, Radosław Waśko Yichen Xu, **Lionel Parreaux**†. 2022. A case for DOT: theoretical foundations for objects with pattern matching and GADT-style reasoning. Proc. ACM Program. Lang. 6, **OOPSLA**2, Article 179 (October 2022), 30 pages. 10.1145/3563342
  († corresponding author)
- **Lionel Parreaux**. 2020. The simple essence of algebraic subtyping: principal type inference with subtyping made easy (functional pearl). Proc. ACM Program. Lang. 4, **ICFP**, Article 124 (August 2020), 28 pages. 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**). 10.4230/LIPIcs.ECOOP.2019.25

- **Lionel Parreaux**, Antoine Voizard, Amir Shaikhha, and Christoph E. Koch. 2018. Unifying analytic and statically-typed quasiquotes. Proc. ACM Program. Lang. 2, **POPL**, Article 13 (January 2018), 33 pages. 10.1145/3158101
- 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**). 10.1145/2882903.2915244

Other Refereed Publications

- ⋆ *Cunyuan Gao* and **Lionel Parreaux**. 2024. Seamless Scope-Safe Metaprogramming Through Polymorphic Subtype Inference. In Proceedings of the 23rd ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences (**GPCE 2024**).
- ⋆ *Chun Kit Lam* and **Lionel Parreaux**. 2024. Being Lazy When It Counts: Practical Constant-Time Memory Management for Functional Programming. In Functional and Logic Programming: 17th International Symposium, **FLOPS 2024**, Kumamoto, Japan, May 15–17, 2024, Proceedings. Springer-Verlag, Berlin, Heidelberg, 188–216. 10.1007/978-981-97-2300-3_11
- ⋆ Yichen Xu, Aleksander Boruch-Gruszecki, and **Lionel Parreaux**. 2021. Implementing Path-Dependent GADT Reasoning for Scala 3. In Proceedings of the 12th ACM SIGPLAN International Symposium on Scala (**SCALA 2021**). Association for Computing Machinery, New York, NY, USA, 22–32. 10.1145/3486610.3486892
- ⋆ **Lionel Parreaux**. 2021. Comprehending Monoids with Class. In The 18th International Symposium on Database Programming Languages (**DBPL 2021**). Association for Computing Machinery, New York, NY, USA, 17–22. 10.1145/3475726.3475728
- **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**). 10.1145/3425898.3426961
- **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**). 10.1145/3337932.3338813
- **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**). 10.1145/3136040.3136043
  🏆 **Best Paper**
- **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**). 10.1145/3136000.3136005

Supervision

Doctor of Philosophy (PhD)

- Anson Yeung (2025–present) 🏆 **HKPFS**
- Mark (Pui Hei) Ng (2025–present) 🏆 **HKPFS**
- Maxime Mulder (2025–present)
- Heung Tung AU (2024–present)
- Ching Hang MAK (2024–present)
- Chun Yin CHAU (2023–present) 🏆 **HKPFS**
- Cunyuan GAO (2023–present)
- Yijia CHEN (2022–present, **passed his PQE**) 🏆 **HKPFS**
- Chun Kit LAM (2022–present, co-supervised with Amir Goharshady, **passed his PQE**) 🏆 **HKPFS**
- Luyu CHENG (2021–present, **passed his PQE**)

Master of Philosophy (MPhil.)

- Ruqing YANG (2023—2025, **graduated**)
- Andong FAN (2022—2024, **graduated**)
- Ishan BHANUKA (2022—2024, **graduated**)
- Chun Yin CHAU (2021—2023, **graduated**)

Research Assistants and Interns

- Gokul Rajiv (Summer 2025)
- Florent Ferrari (Summer 2025)
- Mohammad Broughani (Summer 2023)
- Yihong Zhang (Summer 2021)

Funding

Early Career Scheme (ECS) — **Hong Kong Research Grant Council (RGC)**
Principal Investigator, project code 26208821, period 2022–2025 — *HKD 690,067*
Optimizing Functional Programs by Building on Optimal Graph Reduction Techniques

RDC Industry Grant (Consultancy) — **M-Labs Limited, Hong Kong**
Principal Investigator, project code 23240380P006, period 2023–2024 — *HKD 509,600*
Research in Programming Languages, Compilers and Distributed Computing

RDC Industry Grant (Studentship) — **M-Labs Limited, Hong Kong**
Principal Investigator, project code 23240780P006, period 2024–2028 — *HKD 1,187,500*
Research in Programming Languages, Compilers and Distributed Computing

Young Researcher Supporting Fund — **Huawei Technologies Co., Ltd.**
Gift funding, period 2024–2027 — *HKD 330,000*
A Lightweight Type System for Scope and Effect Safety with Complete Bidirectional & Boolean-algebraic Type Inference

Technology Transfer

- Ongoing research & development with M-Labs Limited, Hong Kong, building a new compiler for the leading-edge control system for quantum information experiments, which is used all over the world.
  `https://cse.hkust.edu.hk/~parreaux/m-labs/`
- Organization of workshop with Huawei R&D to kickstart collaboration on the OpenHarmony/ArkTS project.
  `https://cse.hkust.edu.hk/~parreaux/post/huawei-taco-workshop-2024/`

Teaching

As one of their core contributions to society, academics should not only accumulate and create knowledge, but they should also distil and disseminate it to a wide audience. My approach to teaching is to provide students with a solid foundation in the theoretical and practical aspects of computer science in a way that is engaging and relevant to their future careers.

During my time at EPFL, I was teacher assistant for 8 distinct courses, for a total of 10 semesters of teacher assistantship. These courses ranged from undergraduate math, programming, and theory of computation, to graduate-level database systems and foundations of software. At HKUST, I designed 3 different courses and taught 6 semesters so far, which all received very positive feedback from the students except for one semester, due to some students being disgruntled by an unusually hard midterm exam.

Teaching Method & Philosophy

For me, teaching is a fundamentally interactive activity. During a teaching session, the flow of information should go both ways: from the teacher to the students for facts, methodologies, and guidance; and from students to the teacher for questions, misunderstandings, and particular learning dispositions, allowing the teacher to identify holes and weaknesses in the students' knowledge and to adapt in consequence. For this reason, I always strive to engage the students in a form of dialogue, asking questions and eliciting participation from them, so that I can make sure they remain on the same page as me, and so that the flow of information is maintained.

In the tutorials I design, students learn about more applied topics that are complementary to the theoretical topics of the lectures using *Problem-based Learning*,[1] an *experiential* approach to learning (*active learning*), whereby students work on exercises in small groups and present their solutions to the class, which fosters a more interactive and engaging learning environment.

I evaluate students using a mix of formative and summative assessments, avoiding to rely solely on exams: significant portions of the final grade in my courses are based on homework assignments and projects. Moreover, my exams are *open-book* and favor ingenuity and creativity rather than rote memorization.

Courses Taught

Course code	Semester	Enrolment	SFQ C/I	Course title
COMP3031	Fall 2024	36	4.11/4.33	Principles of Programming Languages
COMP4121	Spring 2024	20	4.50/4.75	Modern Compiler Construction
COMP3031	Fall 2023	39	3.78/3.67	Principles of Programming Languages
COMP4901U	Spring 2023	26	5.00/5.00	Modern Compiler Construction
COMP3031	Fall 2022	31	4.63/4.75	Principles of Programming Languages
COMP6613E	Spring 2022	21	4.89/4.89	Theory of Types and Programming Languages
COMP4901U	Fall 2021	32	4.60/4.67	Computer Language Processing and Compiler Design

Legend: SFQ C/I stands for the Student Feedback Questionnaire scores of the Course (C) and Instructor (I), adjusted with extreme responses excluded, as per `https://student-survey.hkust.edu.hk/`.

SFQ scores in green are above or at the School of Engineering average (corresponding UG or PG level).

Undergraduate Curriculum Development

- Created the **Modern Compiler Construction** course (COMP4121), which was made official in Spring 2024. Previously, HKUST *did not have a compilers course*, a key classical topic in computer science.
- Completely revamped the **Principles of Programming Languages** course (COMP3031). The old material, taught by different instructors over the years, was severely outdated.

[1] Wurdinger, Scott & Rudolph, Jennifer. (2009). Teaching Practices that Improve Student Learning: Five Experiential Approaches. Journal of Teaching and Learning. 6. 10.22329/jtl.v6i1.505.

Postgraduate Curriculum Development

- Taught the **Theory of Types and Programming Languages** course (COMP6613E).

Other Notable Teaching Activities

- Gave a **guest lecture** in Compiling Techniques course at the **University of Edinburgh** in Spring 2024.
- Supervised Final-Year Project/Thesis (**FYP/T**) – 9 projects, **17 students** in total. 1 **Best FYP** Award.
- Supervised Undergraduate Research Opportunities Program (**UROP**) – **12 projects**.

Service Activities

Professional Service

*(**A-ranked** conferences in **bold**.)*

Program Committee Member: **POPL** 2026, **SIGMOD** 2026, PEPM 2026, JFLA 2026, APLAS 2025, DBPL 2025, **OOPSLA** 2025, GPCE 2024, GPCE 2023, **ICFP** 2023, ML Family Workshop 2022, SPLASH Student Research Competition 2022, PADL 2022, Scala 2021, ICFP Student Research Competition 2021, DBPL 2021.

External Reviewer: MFPS 2022, **PLDI** 2025, **OOPSLA** 2026.

External Examiner: Guillaume Martres, PhD thesis, EPFL 2022.

University Service

Advisor of First-Year Students: 2023–2025.

Departmental Service

Member of UG Committee: 2023–2025.

Member of Thesis Examination Committees: 6 MPhil, 5 PhD.

Recognition

Awards and Distinctions

- (2025) 🏆 **ACM SIGPLAN OOPSLA Distinguished Artifact Award** (less than 10% of accepted artifacts)
- (2024) 🏆 **ACM SIGPLAN OOPSLA Distinguished Paper Award #1** (awarded to 7 of 148 accepted papers)
- (2024) 🏆 **ACM SIGPLAN OOPSLA Distinguished Paper Award #2** (awarded to 7 of 148 accepted papers)
- (2024) 🏆 **ACM SIGPLAN ICFP Distinguished Paper Award** (awarded to top 10% of accepted papers)
- (2023) 🏆 **ECOOP Distinguished Artifact Award** (awarded to top 10% of accepted artifacts)
- (2017) 🏆 **GPCE Best Paper Award**
- (2014) 🏆 **EPFL EDIC PhD program fellowship**

Presentations, Seminars, and Invitations

- (November 2024) *Modular Borrowing Without Ownership or Linear Types.*
  ✨ **Invited talk**, HKU PL Group (The University of Hong Kong).
- (March 2024) *A Lightweight Type System for Scope, Effect, and Memory Safety.*
  ✨ **Invited talk**, Huawei Harmony Workshop (Huawei Hong Kong Research Center).
- (January 2024) *When Subtyping Constraints Liberate: Polymorphic Subtype Inference And Scope Safety.*
  ✨ **Invited talk**, University of Cambridge.

- (January 2024) *A Journey Through Deforestation*.
  ✨ **Guest lecture** in Compiling Techniques course, University of Edinburgh.
- (January 2024) *When Subtyping Constraints Liberate: Polymorphic Subtype Inference And Scope Safety*.
  ✨ **Invited talk**, University of Edinburgh.
- (January 2024) *When Subtyping Constraints Liberate: A Novel Type Inference Approach for First-Class Polymorphism*. Conference talk, POPL.
- (November 2023) *A Type System for Scope Safety*.
  ✨ **Invited talk**, HKU PL Group (The University of Hong Kong).
- (July 2023.) *When Subtyping Constraints Liberate: A Novel Approach to First-Class-Polymorphic Type Inference*.
  ✨ **Invited talk**, Huawei Compilers and Programming Languages Workshop (Huawei Research Hong Kong).
- (December 2022) *MLstruct: Principal Type Inference in a Boolean Algebra of Structural Types*. Conference talk, OOPSLA.
- (September 2022) *The Ultimate Conditional Syntax*. Workshop talk, ML Family Workshop.
- (May 2022) *MLscript: Principal Type Inference in a Boolean Algebra of Structural Types.*.
  ✨ **Invited talk**, JetBrains Research (Programming Languages and Tools Lab).
- (January 2022) *Principal Type Inference for Structural Typing in a Boolean Algebra of Types*.
  ✨ **Distinguished Speaker talk**, Code Generation Knowledge Forum at MathWorks.
- (December 2021) *Designing Efficient Systems with Multi-Stage Programming & Other Type-Safe Metaprogramming Techniques*.
  ✨ **Invited talk**, JetBrains Research (Programming Languages and Tools Lab).
- (November 2021) *Principal Type Inference for Object-Oriented Programming in a Boolean Algebra of Types*. Departmental Seminar talk, HKUST.
- (August 2021) *Comprehending Monoids with Class*. Workshop talk, DBPL.
- (November 2020) *Multi-stage Programming in the Large with Staged Classes*. Conference talk, GPCE.
- (August 2020) *The Simple Essence of Algebraic Subtyping: Principal Type Inference with Subtyping Made Easy*. Conference talk, ICFP.
- (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*. Workshop talk, 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) Invitation to Google Compiler and Programming Language Summit, Munich.
- (2016) Invitation to Google PhD Student Summit on Compiler & Programming Technology, Munich.

Open-Source Research Contributions

- **UCS**: artifact of *The Ultimate Conditional Syntax*
  `https://github.com/hkust-taco/ucs` (★ 6 stars)
- **HMloc**: artifact of *Getting Into The Flow*
  `https://github.com/hkust-taco/hmloc` (★ 7 stars)
- **SuperF**: artifact of *When Subtyping Constraints Liberate*
  `https://github.com/hkust-taco/superf` (★ 15 stars)
- **SuperOOP**: artifact of *super-Charging Object-Oriented Programming*
  `https://github.com/hkust-taco/superoop` (★ 11 stars)
- **MLstruct**: artifact of *Principal Type Inference in a Boolean Algebra of Structural Types*
  `https://github.com/hkust-taco/mlstruct` (★ 39 stars)
- **MLscript**: new programming language
  `https://github.com/hkust-taco/mlscript` (★ 187 stars)
- **Simple-sub**: artifact of *The Simple Essence of Algebraic Subtyping*
  `https://github.com/LPTK/simple-sub` (★ 161 stars)
- **dbStage**: database staging framework prototype
  `https://github.com/epfldata/dbstage` (★ 15 stars)
- **Squid**: metaprogramming framework
  `https://github.com/epfldata/squid` (★ 201 stars)