

# Hector Suzanne

DOB March 25<sup>th</sup> 1996

hector.suzanne@lip6.fr

+33 6 99 47 93 49

## Briefly

- **PhD (Current)** “Resource analysis through typing for functional-imperative languages” under the direction of Pr. Emmanuel Chailloux, *APR* team of the *Laboratoire d’informatique de Paris 6*.
- **Université Paris-Cité** Graduated with a Master’s in Mathematics: “Logic, Mathematics and Foundations of Computer Science (LMFI)”
- **Sorbonne Université** Graduated with a Master’s in Computer Science: “Science and Technologies of Software (STL)” research curriculum.
- **ENS Paris-Saclay** Graduated with a License in Computer Science in 2017.
- **English level C2** certified *Cambridge Advanced Grade A* in 2016

## Doctorate Thesis (Current)

### Static resource analysis for high-level languages

*under the direction of Emmanuel Chailloux, at LIP6.*

Sorbonne Université

*February 2021 – Today*

- Financed thanks to the PhD grant of the *Ecole Doctorale Informatique, Télécommunications et Electronique de Paris* (EDITE)
- Conception and implementation of cost analyses for hardware resource over typed functional-imperative languages *à la* ML. Follows from previous internships at LIP6.
- PhD Defense planned for Q4 2023

*(TODO finish this)*

## Education

### Université de Paris

*M2 Mathematics, LMFI curriculum*

Paris

*Sept. 2020 - Sept. 2021*

- **Mathematics and logic** Set theory: ZFC, cardinal and ordinal hierarchies, set combinatorics; Model theory : compactness theorem, Löwenheim-Skolem theorem, quantifier elimination,  $\aleph_0$ -categoricity; Proof theory : Natural deduction, sequent calculus, cut-elimination, Herbrand's theorem, provability in classical, intuitionistic, and linear logic.
- **Categories and higher-algebra** Category theory: functors, natural transformations, (co)-limits, adjunctions, monads; homotopic algebra, enrichment, simplicial sets, higher categories.
- **Computer Science** Calculability: recursive functions and incompleteness, complexity; Proofs for programming: rewriting systems, logical relations, linear logic and proof nets, co-induction and bisimulation.
- **Sample work :**
  - \* *Programming with Coq*: Verified toy-compiler
  - \* *Higher algebra*: Seminar presentation of “A Type theoretic definition of  $\omega$ -categories”, by S. Mimram et E. Finster

### Sorbonne Université

*Computer Science Master's, STL curriculum*

Paris

*2018 - 2020*

- *Functional languages and type systems* at “Master Parisien de Recherche en Informatique”
- **Compilation & Semantics** Typing and semantics; Polymorphism, subtyping and overloading; Formal proofs and dependent types; Abstract interpretation.
- **Algorithms** Analytic combinatorics and random sampling; Average complexity and probabilistic algorithms; Efficient data structures for large-scale storage and processing.
- **Programming** Software engineering and development workflow; Models et representation of data and ontologies; Component-based programming; Preemptive and cooperative concurrency;  $\pi$ -calculus; Synchronous programming; Client-server architecture.
- **Sample work :**
  - \* *Functional languages and type systems*: Typer-optimiser for System-F with GADT
  - \* *Typing and static analysis* : Static analyser through abstract interpretation
  - \* *End-of-year project*: Prototyping a SwiftUI app for tilings *à la* Escher
  - \* *Research seminar* :  $\lambda$ Prolog implemented in Clojure, with high-order unification

### ENS Paris-Saclay

*Licence in Computer Science*

Cachan

*2016 - 2017*

- Admitted through contest with government grant.
- **Algorithms** Theory of proofs and analyses of algorithms for correction and complexity.
- **Logic** Classical logic and its models, compactness; Turing machines, calculability, complexity, reduction techniques; Verification with *Coq*.
- **Semantics** Introduction to formal semantic; Panorama of programming paradigms and their implications;  $\lambda$ -calculus.
- **Programming** Pure functional programming ; Using categorical constructs; UNIX programming ; Fundamentals of networking.

## Past experience

### Teacher's assistant

*Taught 3 programming classes*

Sorbonne Université

*2021 – Aujourd'hui*

- **Introduction to programming with Python:** Co-responsibility of a group of student; lead work groups; created and graded exams.
- **Intro to functional programming with OCaml:** Intensive, project based, 2-week class for double-major student.
- **Intro to functional programming with OCaml (again):** Assisting students during their first encounter with functional-typed languages during work sessions

### Internships at LIP6

*under Pr. Emmanuel Chailloux*

Sorbonne Université

*March-Sept. 2020 & March-Sept. 2021*

- Conception of a memory-cost static analysis for synchronous functional programming following the work of *Dr. Steven Varoumas*
- Extensions for the *OmicroB* and *OCaLustre* platforms for micro-controller programming
- Application to safe embedded software

### Summer Internships at LIP6 and IRILL

*under Pr. Emmanuel Chailloux*

Sorbonne Université

*juin-septembre 2019*

- Conception of a type system for safe concurrency in Cython
- Case study: a formalized account of the Rust borrow-checker in Coq
- Interoperable high-level and low-level programming

### Personal Tutor

*Self-employed*

Tours

*2017-2018*

- Year-long activity giving personalized support in mathematics and computer science
- Undergrads in STEM and economics majors, and high-school students
- Semester-long support and intensive exam prep

### Stage d'été au LIFAT

*under Pr. Denis Maurel*

Université François Rabelais

*juin-août 2017*

- Software engineering for Natural language processing
- Onboarded as a junior developer in a research team
- Contributions to the *Unitex* open-source project

### End of cursus project

*under Michel Bouthemy*

Lycée Descartes

*2016*

- Implementing an algebra engine on finite fields  $\mathbb{F}_{p^n}$
- Modular and purely functional Haskell programming
- Extension to *Reed-Solomon* correcting codes
- Study of the industrial application to large-scale data archival at Facebook