

### Education

- 2020 – 2023 **PhD studies**, *Institut de Recherche en Informatique Fondamentale (IRIF)*, Paris (France)  
Formalisation and implementation of a set-theoretic type system for dynamic languages such as Python or Javascript. Defense planned in 2024.
- 2016 – 2020 **Graduation and Master degree**, *Ecole Normale Supérieure Paris-Saclay*, Cachan (France)  
Graduated in computer science in 2017 (with honors, ranked 2th over 28 students).  
Master degree obtained in 2019: Algorithm and Foundations of Programming (with honors).  
Main courses: specification & verification, programming languages & typing systems, combinatorics & probabilities.
- 2014 – 2016 **Preparatory classes for the Grandes Ecoles**, *Lycée Champollion*, Grenoble (France)  
Main subjects: Math, Physics and Computer Science.
- 2014 **Baccalaureate S option Engineering Science**, *Lycée Paul-Héroult*, Saint-Jean de Maurienne (France)

### Experience

- 2020 – 2023 **Teaching for undergraduate students**, *Université Paris-Cité*, Paris (France)  
Teaching of practical activities and tutorial groups to classes of  $\sim 40$  undergraduate students ( $3 \times 64$ h).
- Summer 2022 **Orgeon Programming Languages Summer School**, *University of Oregon*, Eugene (USA)  
Participation to the OPLSS. During 2 weeks, lectures were given by experts in the programming language area.
- October 2019 **SAT-based model-checking (internship, 10 months)**, *University of Iowa*, Iowa City (USA)  
– July 2020 Implementation of methods aiming to improve the compositional safety analysis within the SAT-based model-checker Kind2: computation of Minimal Inductive Validity Cores, Minimal Cut Sets, etc. Advisors: Cesare Tinelli.
- March 2019 **Set-theoretic type systems (internship, 20 weeks)**, *LRI, Université Paris-Saclay*, Paris (France)  
– July 2019 Formalization of type inference for a language with subtyping and intersection types. Study of some possible extensions for gradual typing and polymorphism. Advisors: Kim Nguyen and Giuseppe Castagna.
- March 2018 **Invariant synthesis (internship, 5 months)**, *Cylab, Carnegie Mellon University*, Pittsburgh  
– July 2018 Automatic synthesis of inductive invariants by counterexample generalization for the Ivy language. Ivy is a language providing interactive tools to easily prove distributed systems. Different decidable fragments of first order logic can be used for specifications and invariants, like the Bernays-Schönfinkel class. Advisor: Bryan Parno.
- Summer 2017 **Static analysis of models (internship, 8 weeks)**, *Carnegie Mellon University*, Pittsburgh (USA)  
Counterfactual causal analysis of Kappa models. Kappa is a rule-based language used for modelling cellular signaling. My goal was to generate causality graphs regrouping events from different simulations. Advisor: Jean Yang.

### Publications

- January 2022 **On Type-Cases, Union Elimination and Occurrence Typing**, *POPL 2022*, conference paper  
Giuseppe Castagna, Mickaël Laurent, Kim Nguyen, and Matthew Lutze. 2022. On type-cases, union elimination, and occurrence typing. Proceedings of the ACM on Programming Languages 6, POPL (January 2022), 1–31. <https://doi.org/10.1145/3498674>  
We extend classic union and intersection type systems with a type-case construction and show that the combination of the union elimination rule and the typing rules for type-cases encompasses occurrence typing.
- January 2022 **Prototype of Typechecker**, *Artifact Digital Object Group*, software artifact  
Giuseppe Castagna, Mickaël Laurent, Kim Nguyen, and Matthew Lutze. 2022. Prototype Typechecker for the Article “On Type-Cases, Union Elimination, and Occurrence Typing.” Association for Computing Machinery (ACM). <https://doi.org/10.1145/3462306>
- 2021 – 2022 **Revisiting occurrence typing**, *Science of Computer Programming*, journal article  
Giuseppe Castagna, Victor Lanvin, Mickaël Laurent, and Kim Nguyen. 2022. Revisiting occurrence typing. Science of Computer Programming 217, (May 2022), 102781. <https://doi.org/10.1016/j.scico.2022.102781>  
We revisit occurrence typing, a technique to refine the type of variables occurring in type-cases.
- August 2021 **Merit and Blame Assignment with Kind 2**, *Formal Methods for Industrial Critical Systems*  
Daniel Larraz, Mickaël Laurent, and Cesare Tinelli. 2021. Merit and Blame Assignment with Kind 2. In Formal Methods for Industrial Critical Systems. Springer International Publishing, 212–220. [https://doi.org/10.1007/978-3-030-85248-1\\_14](https://doi.org/10.1007/978-3-030-85248-1_14)  
Introduction of two new major features of the model checker Kind 2 which provide traceability information between specification and design elements such as assumptions or guarantees in synchronous reactive system models.