Andrea Laretto

iwilare.com | iwilare@gmail.com | iwilare | iwilare

Final-year PhD student working on directed type theory. I focus on designing a simple, concrete syntax of directed type theory to give a logical meaning to theorems in category theory. I'm interested in teaching and implementation of proof assistants, with the ultimate goal of advancing the frontier of PL research and bringing formal methods to everyone.

Education

Early Stage Researcher, supervised by Niccolò Veltri and Fosco Loregian — Tallinn University of Technology, EE
Logic and Semantics Group, High-Assurance Software Lab., Dept. of Software Science — *Jan. 2023 – Jan. 2027*

M.Sc in Computer Science (110/110 hons., avg. 31.48/30) — University of Pisa, IT
"Software: Programming, Principles and Techniques" Curriculum (in English) — *Sept. 2020 – Oct. 2022*

B.Sc in Computer Science (110/110 hons., avg. 29.75/30) — University of Turin, IT
"Languages and Systems" Curriculum (in Italian) — *Sept. 2017 – June 2020*

Publications and Preprints

***Di- is for Directed:* First-Order Directed Type Theory via Dinaturality** (PDF) (code) — POPL 2026
Andrea Laretto, Fosco Loregian, Niccolò Veltri — *Nov. 2025*

Directed First-Order Logic (PDF) — arXiv/2504.11225
Andrea Laretto, Fosco Loregian, Niccolò Veltri — *Jun. 2025*

Counterpart-based Quantified Temporal Logics (PDF) (code) — JLAMP 2025
Fabio Gadducci, Andrea Laretto, Davide Trotta — *Aug. 2025*

Specification and Verification of a Linear-Time Temporal Logic for Graph Transformation (PDF) (code) — ICGT 2023
Fabio Gadducci, Andrea Laretto, Davide Trotta — *May. 2023*

The semibicategory of Moore automata (PDF) (code) — arXiv/2305.00272
Guido Boccali, Bojana Femić, Andrea Laretto, Fosco Loregian, Stefano Luneia — *Apr. 2023*

Completeness for Categories of Generalized Automata ((Co)algebraic pearls) (PDF) (code) — CALCO 2023
Guido Boccali, Andrea Laretto, Fosco Loregian, Stefano Luneia — *Mar. 2023*

Bicategories of Automata, Automata in Bicategories (PDF) (code) — ACT 2023
Guido Boccali, Andrea Laretto, Fosco Loregian, Stefano Luneia — *Mar. 2023*

Theses

Counterpart Semantics for Quantified Temporal Logics: Sets, Categories and Agda (PDF) (code) — University of Pisa, IT
M.Sc Thesis, supervisor Fabio Gadducci, co-supervisor Davide Trotta — *Feb. 2022 – Sept. 2022*

Formalizations of the Church-Rosser Theorem in Agda (PDF) (code) — University of Turin, IT
B.Sc Thesis, supervisor Ugo de' Liguoro, co-supervisor Riccardo Treglia — *Nov. 2019 – Apr. 2020*

Achievements

AILA 3+2 prize for best Italian theses in logic — 2021
Awarded to the B.Sc thesis "Formalizations of the Church-Rosser Theorem in Agda" — *Announcement (in Italian)*

Work and Teaching Experience

Teacher for "Introduction to Category Theory and its Applications" (code) — Tallinn U. of Technology
Teacher for the ITI9200 course at TalTech, "Computer Science" Track — *Feb. 2025 – June 2025*

Teaching Assistant for "Introduction to Category Theory and its Applications" (code) — Tallinn U. of Technology
TA for the ITI9200 course at TalTech — *Feb. 2024 – June 2024*

Teaching Assistant for "Functional Programming" — Tallinn U. of Technology
TA for the "Functional Programming" ITI0212 course at TalTech — *Feb. 2023 – June 2023*

Doctoral Tutor — Tallinn U. of Technology
Private tutor for PhD students on type theory and category theory — *Aug. 2025 – Dec. 2025*

University Tutor — Turin
Private in-person Haskell and Agda tutoring with CS university students — *Feb. 2022 – July 2022*

Upwork Tutor — online
Remote assistance with Haskell and OCaml homeworks, projects, university exams — *May 2021 – Sept. 2021*

Project work (selection)

Contributions to *agda-categories* — Nov 2021 – ongoing
Additions to the *agda-categories* library; coEilenberg-Moore categories and Mac Lane comparison functor, Kleisli extension and isomorphisms in Kleisli, simple/ordinary slices and Kleisli/Eilenberg-Moore categories of the product comonad
agda/agda-categories

Formal Methods in Agda — Sept. 2020
Agda formalizations of material in the *"Formal Methods for Computer Science"* bachelor's course: semantics of imperative languages, separation logic and frame rule, Hoare logic, security-based type systems with their type preservation and progress
formal-methods

***MicroC* LLVM compiler** — Dec. 2020 – Feb. 2021
Compiler for a C-like language written in OCaml, using LLVM as compilation backend and ocamllex/Menhir as frontend; support for multidimensional arrays and structs
compiler-course-unipi

***Monoid* Forth** — Aug. 2021
Bootstrapping x86_64 operating system and minimal Forth interpreter, easily portable and self-bootstrapping, with small-footprint UEFI interfacing and support
monoid-forth

***Sol* language** — Nov. 2019
Toy interpreter for a Smalltalk-inspired programming language written in Java, with metaclasses, tail recursion, dictionary-based class reflection, HTTP and sockets support
sol-lang

Video Motion Detection — June 2022
Parallel application in C++ for a convolution-based motion detector, providing threads, FastFlow, and OpenMP implementations and their performance analysis. Project work for the *"Parallel and Distributed Systems: Paradigms and Models"* master course
video-motion-detection

Interests

- type theory, categorical logic
- implementation of dependent type theory, proof assistants, homotopy type theory
- models of computation, λ-calculus, confluence, term rewriting, graph rewriting, e-graphs
- functional programming, programming language theory, operational and denotational semantics
- compilers, abstract machines, static analysis, concatenative programming

Technical Skills

Languages: Haskell, Agda, OCaml, Rust, TypeScript, Idris, Elm, Scheme, C/C++, Java, Python, JavaScript, HTML/CSS
Frameworks: LLVM, Menhir, React, Dune, Parsec, Megaparsec, Warp, NumPy, OpenMP, Pandas, SciPy, Matplotlib
Tools: Visual Studio Code, Git, GitHub, LaTeX, TikZ, Typst
Operating Systems: NixOS, Windows, Xubuntu

Languages

Italian: native
English: professional (C1 self-assessed, B2 certificate)
Japanese: good reading skills, limited working proficiency
Russian: beginner