Andrea Laretto

iwilare.com | iwilare@gmail.com, andrea.laretto@taltech.ee | iwilare | iwilare

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 — *Nov. 2022 – Nov. 2026*

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 — *Sept. 2017 – June 2020*

Talks

Semantics for Counterpart-based Temporal Logics — Tallinn, EE
World Logic Day 2023 - Logic in Estonia Workshop — *14th Jan. 2023*

Categorical Semantics for Counterpart-based Temporal Logics in Agda — Pisa, IT
3rd ItaCa Workshop — *21st Dec. 2023*

Categorical Semantics for Counterpart-based Temporal Logics in Agda — Bergen, NO
Nordic Workshop on Programming Theory 2022 — *2nd Nov. 2022*

Publications and Preprints

Specification and verification of a linear-time logic for graph transformation — arXiv/2305.03832
Fabio Gadducci, Andrea Laretto, Davide Trotta — *May. 2023*

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

Completeness for categories of generalized automata (Coalgebraic Pearl) — CALCO 2023
Guido Boccali, Andrea Laretto, Fosco Loregian, Stefano Luneia — *Mar. 2023*

Bicategories of automata, automata in bicategories — arXiv/2303.03865
Guido Boccali, Andrea Laretto, Fosco Loregian, Stefano Luneia — *Mar. 2023*

Theses

Counterpart Semantics for Quantified Temporal Logics: Sets, Categories and Agda — 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 — 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

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

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, with Fosco Loregian *(@tetrapharmakon)*; monad morphisms, category of adjunctions splitting a monad, representable profunctors, coEilenberg-Moore categories and Mac Lane comparison functor, Kleisli extension and isomorphisms in Kleisli, quantales, simple/ordinary slices and Kleisli/Eilenberg-Moore categories of the product comonad
agda/agda-categories

Formal Methods in Agda — *Sept. 2020*
Agda formalizations for some of the 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; supporting 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

Interests

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

Courses attended (selection)

Foundation of Computing — *2021*
Category theory, algebraic and logical foundations; higher-order, recursive typing, Curry-Howard, CCC; petri nets, PCF, π-calculus and their models; LTS and coalgebras
Ugo Montanari
Grade: *30L/30*

Principles for Software Composition — *2021*
Models of computations, operational and denotational semantics; modelling languages with higher-order, concurrent, probabilistic features; temporal and modal logics
Roberto Bruni
Grade: *30L/30*

Languages, Compilers, and Interpreters — *2020*
Lexical analysis, parsing, intermediate representations, abstract interpretation; laboratory project developing a C-like compiler in OCaml with LLVM, ocamllex, Menhir
Letterio Galletta, Roberta Gori
Grade: *30L/30*

Advanced Programming — *2020*
Modern concepts and pragmatics of programming languages; OOP and design patterns, JVM and Java Streams, Haskell and monads, implementation and semantics of Python
Andrea Corradini
Grade: *30L/30*

Software Validation and Verification — *2022*
Temporal and modal logics, LTL, CTL, CTL*; safety and liveness properties, fairness; Büchi automata, ω-regular properties, model checking algorithms; spatial logics
Fabio Gadducci
Grade: *30/30*

Laboratory for Innovative Software — *2022*
Hands-on research work on hardware-based security; student group project implementing microarchitectural CPU models with secure interruptible enclaves in OCaml
Gian-Luigi Ferrari, Chiara Bodei
Grade: *30L/30*

Language-based Technology for Security — *2022*
Low-level security flaws, memory corruption, language implementation and attacks; security measures, hidden channels, information flow security, static analysis
Gian-Luigi Ferrari, Chiara Bodei
Grade: *30L/30*

Formal Methods for Computer Science *2019*

Hoare logic, imperative semantics for IMP, Isabelle theorem prover (now in Agda!); type systems for confidentiality and information flow, separation logic and VeriFast

Ugo de' Liguoro
Grade: *30L/30*

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
Operating Systems: NixOS, Windows, Xubuntu

Languages

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

Short self-introduction

Always eager to learn new concepts, with equal interest in both the theoretical and the practical point of view. Extremely passionate about teaching and explaining new concepts to others, striving for practicality, simplicity, and ease of understanding. Comfortable working both in groups as well as delivering results individually. Currently a mentee in the SIGPLAN-M long term mentoring program under the supervision of Fosco Loregian.