Jordan ISCHARD

📍 Orléans, France ✉ jordan.ischard@univ-orleans.fr 📱 06 45 81 02 10

RESEARCH TOPICS

My PhD focuses on the verification of reactive programming languages and, more specifically, on functional reactive programming (FRP) languages with imperative features. I am interested in the FRP languages, like `Yampa` which is purely functional, and their extension with side effects or real parallelism. I use a proof assistant called Coq/Rocq to guarantee my results. I also work on the transformation of parallel programs verified in this same proof assistant.

Keywords: Formal methods, verification, language semantics, reactive programming, functional programming, proof assistant

EDUCATION

PhD in Computer Science since 2021
University of Orléans, Orléans, France
Team: LMV team, LIFO laboratory
Subject: Semantics analysis of reactive programming languages using modal and linear logic

Master's degree in Computer Science 2019-2021
University of Orléans, Orléans, France
Related coursework: Android application, website and geographic information system development
Rank: 2/43

Bachelor's degree in Computer Science 2016-2019
University of Orléans, Orléans, France
Related coursework: algorithmic, `C/C++`, `Java`, `OCaml`, network, graph theory, automata and framework web

RESEARCH INTERNSHIPS

Formalization of a mathematical theorem 2021 (6 months)
LMV team, LIFO, University of Orléans, Orléans
Formalization in `Coq` of abstract simplicial complexes applied on weak memory models.

- ✓ `Coq` proof assistant training
- ✓ Modelization of abstract simplexes and complexes

Implementation of a functional kernel 2019 (5 months)
Subject: Synchronous reactive programming and implementation of an abstract machine for a synchronous reactive functional kernel in `OCaml`

- ✓ Implementation of an abstract machine named `SECD`
- ✓ Extension of the abstract machine in order to handle signals

EXPERIENCE

Support for research engineer since October 2024

LMV team, LIFO, University of Orléans, Orléans

Subject: Research and development with the SyDPaCC Framework for the development of verified functional parallel programs based on simple and clear functional specifications.

Temporary research assistant (BIATSS) January 2020 - June 2020

LMV team, LIFO, University of Orléans, Orléans

Subject: Implementation of functional kernel with synchronous features in `OCaml`

- ✓ Improvements to extensions suggested during the bachelor internship
- ✓ Creation of a top-level loop with `dune`

PUBLICATIONS

Articles

- ✓ Frédéric Loulergue and Jordan Ischard (2024, October). *SyDPaCC: A Framework for the Development of Verified Scalable Parallel Functional Programs*. In International Symposium on Leveraging Applications of Formal Methods (pp. 274-295). Cham: Springer Nature Switzerland.
- ✓ Jordan Ischard, Frédéric Dabrowski, Jules Chouquet and Frédéric Loulergue (2025, March). *A Mechanized Formalization of an FRP Language with Effects*. In ACM Symposium on Applied Computing (SAC).

Preprint(s)

- ✓ Frédéric Dabrowski and Jordan Ischard (2025). *Functional Reactive Programming with Effects, A More Permissive Approach*. arXiv preprint arXiv:2503.06162.

PRESENTATIONS

- ✓ *A Mechanized Formalization of an FRP Language with Effects*, Symposium On Applied Computing (SAC), Catania, Sicily (April 2025)
- ✓ *SyDPaCC: A Framework for the Development of Verified Scalable Parallel Functional Programs*, Journées informatique en Région Centre-Val de Loire (JIRC), Tours (November 2024)
- ✓ *SyDPaCC: A Framework for the Development of Verified Scalable Parallel Functional Programs*, ISoLA, Crete (October 2024)
- ✓ *Formalization of an FRP language with references*, SeSTeRce Day, University of Orléans, Orléans, France (September 2023)
- ✓ *An Overview of Reactive programming*, PhD students Day, University of Orléans, Orléans (April 2022)
- ✓ *When the purely functional has effects*, Journées informatique en Région Centre-Val de Loire (JIRC), University of Orléans, Orléans (April 2022)

TEACHING

During my 2-year master's degree, I worked as a tutor for 1st-year bachelor's students for the *Algorithms and programming* class for a period of 4 months. This class covers the basics of algorithms and introduces the language `Python`.

During the first 3 years of my PhD, I carried out complementary teaching assignments (detailed below). For each class, I contributed to the correction of assessments, exams and projects. For the *Functional Programming* class, a contribution to the preparation of assessments was provided, and for the *Imperative and Object-Oriented Programming* class, I took part in the creation of tutorial sheets.

Year	Class	Grade	Seminar	Tutorial
2021-2022	Imperative and Object-Oriented Prog.	L3		48
	Functional Prog.	L2		16
2022-2023	Imperative and Object-Oriented Prog.	L3		48
	Functional Prog.	L2		16
2023-2024	Logic	L1	16	8
	Framework Web	L3		20
	Compilation	M1		18

Logic 16h of seminars and 8h of tutorials

This course provides an introduction to propositional and predicate logic. The proof assistant `Coq` is used during the tutorials. This class is graded throughout the semester via continuous assessments, which are the responsibility of the seminar supervisors, and a final exam.

Framework web 20h of tutorials

The course introduces two frameworks for web programming: `Symfony` and `Angular`.

Compilation 18h of tutorials

This master's-level course breaks down the compilation of a program, starting with the definition of the grammar and ending with translation into the low-level language `assembler`. For this, the ANTLR4 tool is used via the `Java` language. The class is graded by continuous and final assessments.

Imperative and Object-Oriented Programming 48h of tutorials

The aim of this course is to use the `C` and `C++` languages in order to understand what explicit memory management involves. By the end of the course, students will be able to develop in both `C` and `C++`. This class is graded through a mid-term paper exam, a final exam and a project.

Functional Programming 16h of tutorials

This course introduces functional programming via the `OCaml` language. This class is graded through a mid-term paper exam, a final exam and a project.

SKILLS

Computer Languages

Advanced
`Rocq/Coq`, `OCaml`, `C` and `Java`
Intermediary
`Python`, `C++` and `Haskell`
Beginner
`C#`, `PHP`, `SQL` and `TypeScript`

Frameworks

`Symfony`, `Angular`, `API Rest` and `Spring`

Tools

Git, LaTeX, bash, Docker, vscode and IntelliJ

Language

French (mother tongue) and English (B2)