The project

I am interested in computational foundations of mathematics. In a normal day, mathematicians do not need to worry about the fundamental postulates and rules of inference. Even if the reasonings are dressed as formal proofs respecting those principles, it is socially accepted to sweep under the rug as many details as needed to achieve a higher goal: a clear, concise and understandable by humans presentation of the main ideas.

A complete reversal of this situation happens when we try to explain mathematics to a computer. The most insignificant details of our theory can become tedious (but necessary) bureaucracy when proving theorems in a completely formal way. Our usual foundations in set theory are usually not well adapted to computer formalization, and checking mathematics in a computer is not simply a pastime! some mathematical problems have been formalized for the first time by a computer and, in safety-critical software systems where human lives are at stake, formal verification is prerequisite.

Thus, we have a need to revisit the foundations of mathematics to make them easily intelligible for computers; that is, we need our foundations to have a computational interpretation and ensure that working with them in a conceptual level in computers is easy. Fortunately, we have all the necessary tools at hand.

- The *Brouwer-Heyting-Kolmogorov interpretation* showed in the 30s that a certain kind of mathematical logic (constructive or intuitionistic logic) could be interpreted in a computational way.
- *Topoi* were invented by one of the greatest mathematicians of the past century, Alexander Grothendieck, while studying algebraic geometry; later, it was discovered that each of them could be seen as a new world (or model) of these constructive mathematics.
- The Swedish philosopher Per Martin-Löf proposed a *type theory* (slightly different from the usual set-theoretical approach) that could be thought at the same time as a programming language and as a language for these constructivist mathematics. Recently, connections with the fundamental structure of spaces (topology) have been discovered.

What is the goal then? We need to explore these new worlds of mathematics, study how are they related to type theory and start writing mathematics using programming languages. How could this be achieved exactly is a subject of active research. In a not-so-distant future, we could see collaborative libraries of formalized mathematics, an open repository of mathematics where each one of us could submit a proof and have it checked by the computer. These same mathematical proofs could be used while programming, to ensure the correctness of our software. Although this vision is still far, it would be revolutionary both for mathematics and programming.

About me

Five years ago, I chose a double degree in mathematics and computer science; interest in mathematics had been sparked in me while preparing for international mathematical olympiads at high school; and I really liked the logical aspect of designing programming languages, but I had a strong feeling that I was missing the relation they may have with pure mathematics and logic, so my first resolution was to learn about theory of programming languages. Thus, I discovered functional programming, a way of programming closely related to algebraic thinking, type theory, and how they provided a mathematical ground for languages.

At the same time, I started reading about the theory of *topoi* (in general, the branch of mathematics called *category theory*) in an algebra book and the simplicity of the proofs and the generality of the applications captured my interest. When I learnt that they were also related to programming, I decided to devote the majority of the sparse time that I had during my degree to study this relation.

Note that neither category theory nor functional programming were being taught at my university; I decided that the best way to learn would be to teach and discuss with my colleages, so I started organizing weekly seminars on these topics, using free software and writing public notes in Creative Commons to teach. Four years later, I am the coordinator of the vibrant community of students at our university that we have created. We organize seminars, promote the adoption of free software and help students of mathematics and computer science (see https://libreim.github.io/). This experience has taught me many skills I wouldn't have been able to adquire otherwise: how to coordinate a community at a technical and personal level, how to give public talks on the topics I am passionate about while adapting the level to different audiences, how to write mathematics in a didactic way and how to effectively promote the adoption of free software and open knowledge policies at my university.

However, it would be naive to think that I can learn all that I need while staying at a university in which there are no courses on these topics. I have studied precisely the necessary tools to work on type theory and foundations of mathematics, and now I would like to finish this formation and adquire the relevant background (in Logic, Category theory, Type theory and Foundations of mathematics) to pursue a research career with the project I described in the first section in mind.

Master programmes in Logic and Mathematics

The Netherlands has a long tradition in constructive mathematics thanks to Brower's intuitionism and the school that his academic descendants established; more recently continued by topos theorists such as Ieke Moerdijk. Additionally, is has a great computer science heritage with Dijkstra and the first universities that advocated a scientific approach to programming.

This translates to the fact that Dutch universities offer the possibility of a specialization in category theory and logic and courses on topos theory, type theory and logic through the the national Mastermath programme, shared between multiple universities. Explicitly, I am interested in

- the Master in Logic at the University of Amsterdam,
- the Master in Mathematical Sciences at the University of Utrecht, and
- the Master in Mathematical Foundations of Computer Science at the Radboud University Nijmegen.

My intention is to pursue a master degree with a research career in mind; although I would not be closed to other opportunities, I have enjoyed my first experiences of mathematical research thanks to a introduction to research grant for undergraduate students.