Contribution ID: **163** Type: **Talk (15min + 5min)**

On Embedding Code Extracted From Coq Formalisations into Data Analysis Workflows

Wednesday 26 February 2025 11:00 (20 minutes)

Formal methods are essential for ensuring the correctness of algorithms and they are used in many different variants. Algorithms are verified using temporal logic, separation logic and type-theoretic approaches, for example.
In particular, type-theoretic formalisms, as implemented by the Coq Proof Assistant offer the possibility of even synthesising software from its specification by proving the existence of a function that satisfies the specification.
And while these methods are commonly used in security and safety-critical applications, such as operating systems, cryptography, and network communications, their use in data analysis workflows appears to be rather limited.
However, in the analysis of scientific data, the correctness of the results is clearly desirable, if not essential.
Furthermore, proof assistants such as Coq are much less constrained by the architecture, and can, for example, compute on integral values of arbitrary length, thus providing much higher precision than general-purpose programming languages.
Finally, the Coq ecosystem provides many proven correct algorithms, some of which could be reused as part of research software, e.g. for data analysis tasks.

In this talk we present the advances in the Coq software extraction mechanism, namely the ability to generate foreign function interfaces for communication between extracted OCaml code and plain C code.
We focus on improvements in both the type safety of the data exchange and the maintainability of the generated interfaces. We discuss a simple example application for data analysis implemented in Coq. We then show how this application, including the foreign function interface, can be extracted and integrated into a sample data analysis workflow.

I want to participate in the youngRSE prize

yes

Primary author: FRANK, Mario

Presenter: FRANK, Mario

Session Classification: Workflows for data pipelines

Track Classification: Data and Software Management: computational workflows