Graphical Reasoning, in Coq

Gabriel RADANNE – Inria CASH/LIP
Yannick ZAKOWSKI – Inria CASH/LIP

2022-2023

1 Context

« Draw a picture ! »— you have probably heard this sentence come out of your math teacher's mouth thousands of time ! Graphical representations are indeed a powerful tool to abstract away and ease intuition when dealing with complex constructions and proofs. Most obviously, geometrical reasoning benefits from an Euclidian perspective when one gets lost in coordinate-based computations [2]. But other domains draw similar benefits. In computer science, finite state automata are routinely represented graphically to provide intuition [4]. Diagrammatic reasoning has been the method of choice used by quantum computer scientists to reason about their computations [3, 1]. The Penrose project [5] is probably among those pushing the art of mathematical visualization the furthest !

But all these approaches promise more than a mere intermediate uncluttered visualization : the pictures themselves may form a mathematical language upon which proofs can be conducted ! Moving away from cute informal « proofs » of elementary algebraic equations, rigorous foundations to these graphical languages has been built, legitimizing them more and more as a proper way to prove theorems.

Far away from intuition, the working mathematician may also be concerned with trust in a proof. To this end, it is increasingly more common to formalize one's proofs in a proof assistant, such as Coq for instance. By nature, the degree of details involved in a machine-checked proof is tenfold. As a consequence, it gets even easier to get lost in the minute details of the formalism at hand. This hence begs the question : could we draw inspiration from the previously mentioned literature in order to be able to summon via a new Coq command a graphical representation of the current goal [2] ? Looking one step further : now that solid foundations exist to describe legal operations that one can perform on visualizations, could we interact with the visualisation, and send back on the Coq side an open proof term, or a sequence of tactics corresponding to the interactions performed ?

2 Internship objective

The goal of this internship is to develop a Coq plugin for graphical visualization and graphical reasoning over carefully selected abstractions. It will be split into two axes of work.

On the theoretical side, we aim to identify specific mathematical domains (adequate algebraic structures, Euclidian geometry, Floyd-style program verification, ...) suitable for graphical reasoning. A potential starting point would be to consider the diagrammatic approach applied to the particular case of linear algebra [3]. Other domains can be considered, depending on the candidate's interests.

On the practical side, the goal of this internship is to :

— develop the Coq theory required to manipulate the algebraic structure identified in the theoretical step ;
— develop a Coq plugin to interface this theory with an interactive graphical representation ;
— design a mechanism for interacting graphically with the representation, and deriving Coq-level information about this manipulation.

Both aspects will be pursued in concert and will be developed according to the taste of the candidate.

3 Internship

The internship will take place in the CASH Team, LIP lab, in Lyon France.

Candidate profile The candidate should ideally be familiar with formal reasoning in logic and algebra. From the practical point of view, a basic experience in software programming in OCaml and usage of collaborative tools such that `git` is mandatory.

Références

[1] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes : A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017. doi : 10.1017/9781316219317.

[2] Julien Narboux. A graphical user interface for formal proofs in geometry. *Journal of Automated Reasoning*, 39, 08 2007. doi : 10.1007/s10817-007-9071-4.

[3] Pawel Sobocinski. Graphical linear algebra. https://graphicallinearalgebra.net/, 2015.

[4] Robin Piedeleu and Fabio Zanasi. *A String Diagrammatic Axiomatisation of Finite-State Automata*, pages 469–489. 03 2021. ISBN 978-3-030-71994-4. doi : 10.1007/978-3-030-71995-1_24.

[5] Katherine Ye, Wode Ni, Max Krieger, Dor Ma'ayan, Jenna Wise, Jonathan Aldrich, Jonathan Sunshine, and Keenan Crane. Penrose : From mathematical notation to beautiful diagrams. *ACM Trans. Graph.*, 39(4), 2020.