Experiments with computable matrices in the Coq system

Maxime DÉNÈS, Yves BERTOT

INRIA Sophia-Antipolis Méditerranée
Maxime.Denes@inria.fr, Yves.Bertot@inria.fr

Matrices and vectors are a traditional example of how the formalization of mathematics can benefit from dependent types. However, usual solutions turn out to be unwieldy in practice, because dependent types impose too many constraints on programming.

The math-components library proposes a two-facet approach, where dependent types can be forgotten at the level of programming and recovered to capture dimension constraints for matrices and vectors. Thanks to this approach, a lot of properties of matrices can be and have been formalized, in work prior to ours [2, 3].

However, the matrices as they are described in the math-components library are fine-tuned for proofs and the execution aspect is secondary. To provide reasonably efficient computations on matrices it seems natural to describe again the algorithms on simply typed data and prove their correctness with respect to the mathematical description. We will comment on this approach and the techniques to relate algorithms and formal descriptions.

In a third stage, we can improve efficiency by moving away from the traditional approach of functional programming based on lists and considering datatypes whose complexity is closer to arrays [1]. This requires a reflection on the internal machinery of the Coq system but can bring important benefits in feasibility of computations and, hopefully, proofs.

We will illustrate these techniques on fundamental algorithms like determinant computations or fast matrix product.

References

[1] M. Armand et al. “Extending Coq with imperative features and its application to SAT verification”. In: *Interactive Theorem Proving* (2010), pp. 83–98.

[2] Y. Bertot et al. “Canonical big operators”. In: *Theorem Proving in Higher Order Logics* (2008), pp. 86–101.

[3] S. Biha. “Finite groups representation theory with Coq”. In: *Intelligent Computer Mathematics* (2009), pp. 438–452.