Proposal MAP 2011

Thierry Coquand, Henri Lombardi, Marie-Francoise Roy and Bas Spitters*

November 28 – December 2, 2011

Abstract

This is a proposal for the MAP workshop to be held at the Lorentz center in Leiden. MAP[1] is an acronym for Mathematics: Algorithms and Proofs. The MAP community meets yearly and has been at the Lorentz center before in 2007.

This meeting will be combined with a meeting of the ForMath[2] EU STREP-project on the formalization of mathematics.

1 Scientific case and motivation

The goal of the seminar is to bring together people from the communities of formal proofs, constructive mathematics and computer algebra (in a wide meaning). This is a continuation of previous meetings in Schloss Dagstuhl (2003 and 2005), Luminy (2004), Castro Urdiales (2006), Leiden (2007), Trieste (2008), Monastir (2009), Logroño (2010). The previous meetings have been quite successful in bringing these communities closer together. The European ForMath project which deals with the computer verified development, of both proofs and algorithms, in group theory, algebraic topology, linear algebra and computational analysis, may be seen as an outgrowth of the previous meetings.

One objective of the workshop is to bridge the gap between conceptual (abstract) and computational (constructive) mathematics, by providing a computational understanding of abstract mathematics. The connections between topos theory and dynamical proofs and evaluation and the dawning connections between type theory (from logic and functional programming), homotopy theory and higher category theory form bridges over this gap. In this way, it is becoming increasingly clear that many parts of abstract mathematics can be made constructive and even computational and that abstract mathematical techniques contain an underlying constructive content. We are not only interested in correct algorithms however, but also in the mathematical clarity that these concrete presentations provide.

Needless to say, computer algebraists want and need correct programs. The current generation of programming languages does not always make this correctness transparent. Constructive algebra provides a convenient framework for specifying the behaviour of e.g. the `homalg` computer algebra system for homological algebra. Experiments with a verified implementation of parts of Kenzo in Coq's type theory have started. Such an implementation includes a machine verified correctness proof. Conversely, computer algebra algorithms serve both as an important test for formal proofs and as a source of automation.

Real algebraic geometry is a relevant instance of such a situation since real algebraic numbers can be dealt with either algebraically or by well controlled approximations. When such algorithms are correctly controlled, they actually deal with real and complex numbers in the constructive meaning of these objects. So computer algebra fills many objectives of computational analysis. Conversely, (exact) numerical computations can often be used as decision procedures.

In numerical analysis, the danger of making errors in floating point computations are well recognized. Exact numerical algorithms allow one to shift the burden of ensuring the correctness from

* *Email:* `spitters@cs.ru.nl`

[1] `http://map.disi.unige.it/`

[2] `http://www.tinyurl.com/formath`

the user to the machine. These exact numerical algorithms are best described by constructive logic. Computational homology is one of the most active applications of exact numerical computations.

1.1 Timeliness

The ForMath project, which started in 2010, shows that verified programming has matured enough to start to become interesting for both computer algebra and the verification of huge (computer) proofs in mathematics. It is crucial to obtain input from the general MAP-community. As an example, computational homology applies exact computational analysis and homology to hard analytical problems, in e.g. differential equations. In this way, we arrive at a proof by rigorous computer computations, which lack a short pen-and-paper proof. In the future we would like the underlying algorithms, and their implementation, to be formally verified, as we cannot simply verify the outcome of the computations. The ForMath project works towards such a future by working on the verification of computational analysis and algebraic topology. The tutorial by Escardo and Simpson will focus on the first topic. There will be contributed talks on computational algebraic topology.

After finishing the proof of the Bloch-Kato conjecture, Fields medalist Voevodsky[3], became 'convinced that the most interesting and important directions in current mathematics are the ones related to the transition into a new era which will be characterized by the widespread use of automated tools for proof construction and verification.' We have two invited speakers (Avigad, Gonthier) on proof assistants. Gonthier's group aims to formally verify the Feit-Thompson theorem, the 300 page proof which constitutes a main part of the classification of finite simple groups. To find a (the?) right language for the formal development of mathematics, Voevodsky and Awodey have been developing the connections between homotopy theory, type theory and higher categories. A main research goal is to find a computational interpretation of Voevodsky's univalence axiom. In 2012, the Princeton Institute of Advance Studies will host a year long program on this topic[4]. There is a substantial overlap with the aims of the MAP community. Hyland and Streicher will lecture on these connections.

The book 'Algèbre Commutative, Méthodes constructives (Modules projectifs de type fini)' by Lombardi and Quitté (2010) shows that commutative algebra can be presented in an entirely constructive way and with attention to practical algorithms without sacrificing the beauty of the presentation. Such a presentation forms a crucial base for the (verified) development of a computer algebra and is thus of crucial importance as it unities all aspects of the work in the MAP community. As a concrete example, Thierry Coquand, Henri Lombardi and Peter Schuster are investigating schemes in formal topology, a computational presentation of topology based on Grothendieck's sites, while at the same time Mohamed Barakat is exploring the representation of coherent sheaves over projective schemes in the `homalg` computer algebra package. Surprisingly, sheaves also allow one to apply *constructive* commutative algebra to non-commutative algebra in the foundations of quantum theory. This is recent work by Heunen, Landsman and Spitters. Finally, sheaf theory has been implicit in the work on dynamical proofs. Coquand's work (2010) connects this to sheaf models of type theory. The MAP meeting is the ideal place to develop and disseminate these ideas.

As a final connection between core logic and practical verification we mention the question raised by, for instance, Hales verification of the Kepler conjecture. The verification of this huge computer proof, which is well on its way, is spread out over several provers: Coq and Isabelle/HOL. This raises the question of interoperability between these systems. In particular, it prompts us to understand and develop the connections between higher order logic from topos theory (used in HOL) and type theory (used in Coq).

[3] `http://www.math.ias.edu/~vladimir/Site3/Univalent_Foundations_files/univalent_foundations_project.pdf`

[4] http://www.math.ias.edu/sp/univalent

2 Program

Tutorials

- Mohamed Barakat (Kaiserslautern) and Alban Quadrat (INRIA Saclay) - D-modules
- Alex Simpson (Edinburgh) and Martin Escardo (Birmingham) - Categorical axioms for functional real-number computation

Invited talks

- Jeremy Avigad (CMU)- Proof theory and formalization of mathematics
- Ieke Moerdijk (Utrecht)- Topos theory
- Thomas Streicher (Darmstadt)- Type theory, homotopy and higher categories
- Martin Hyland (Cambridge) - Type theory, homotopy and higher categories
- Georges Gonthier (Microsoft research)- Group theory in Coq