Mathematical Structures in Dependent Type Theory

Assia Mahboubi
Inria, Nantes, France
LS2N, Université de Nantes, France
Vrije Universiteit Amsterdam, The Netherlands
http://people.rennes.inria.fr/Assia.Mahboubi/
Assia.Mahboubi@inria.fr

Abstract

In this talk, we discuss the role and the implementation of mathematical structures in libraries of formalised mathematics in dependent type theory.

2012 ACM Subject Classification Theory of computation → Type theory; Theory of computation → Logic and verification

Keywords and phrases Mathematical structures, formalized mathematics, dependent type theory

Digital Object Identifier 10.4230/LIPIcs.CSL.2021.2

Category Invited Talk

1 Summary

Since the early writings of the Nicolas Bourbaki group [3], mathematical structures are used in all fields of mathematics to structure the mathematical language, its vocabulary and its notational apparatus. An instance of a given structure is a carrier set equipped with some identified elements, with some operations on the carrier, and with some properties – called the axioms of the structure. Put in good use, these abstractions clarify the mathematical discourse for a knowledgeable audience, while emphasising correspondences between seemingly unrelated mathematical objects. Classical model theory provides a mathematical formalisation of the notion of structure [12], of which algebraic structures are an instance.

The past decade has seen the advent of several large-scale libraries of formalised mathematics [6, 2, 9, 17], most of which framed by a hierarchy of formal algebraic structures [10, 13, 8, 17]. The latter hierarchies can be seen as a formal-proof-engineering device, which organises inheritance and sharing in a similar way as the design patterns of object-oriented programming [7, 4]. The implementation and the features of these hierarchies depend both on the flavour of foundations the proof assistant is based on, and on the implementation in the prover of enhanced type inference procedures [15, 11, 16, 1, 14]. The central idea is to take benefit of some form of type inference in order to compute automatically the missing information in the user input, so as to achieve concision in the statement of formal sentences, while still providing a well-formed term to the prover’s checker.

This talk will focus more specifically on the case of formalisations, and proof assistants, based on variants of dependent type theory. This setting allows in particular a first-class representation of structures using dependent tuples (also called telescopes [5]). It will discuss the recent techniques proposed to design these hierarchies, their pitfalls, the corresponding achievements, and their limitations.

References

1 Andrea Asperti, Wilmer Ricciotti, Claudio Sacerdoti Coen, and Enrico Tassi. Hints in unification. In Stefan Berghofer, Tobias Nipkow, Christian Urban, and Makarius Wenzel,
Mathematical Structures in Dependent Type Theory

2:2 Mathematical Structures in Dependent Type Theory


15 Amokrane Saïbi. Typing algorithm in type theory with inheritance. In Peter Lee, Fritz Henglein, and Neil D. Jones, editors, *Conference Record of POPL’97: The 24th ACM SIGPLAN-