

Quotients in Dependent Type Theory

Andrew M. Pitts 

Department of Computer Science & Technology, University of Cambridge, UK

Abstract

Constructs that involve taking a quotient are commonplace in mathematics. Here I will consider how they are treated within dependent type theory. The notion of *quotient type* has its origins in the Nuprl theorem-proving system [4] for extensional type theory. Later Hofmann formulated a version for intensional type theory in his thesis [7]. This depends on having a pre-existing notion of intensional identity type. Hofmann used Martin-Löf's notion, the indexed family inductively generated from proofs of reflexivity [9, chapter 8]. The recent homotopical view of identity in terms of path types [10] gives a more liberal perspective and has brought with it the notion of *higher inductive type* (HIT) [8], subsuming both inductive and quotient types.

Inductively defined indexed families of types (in all their various forms) are perhaps the most useful concept that dependent type theory has contributed to the practice of computer assistance for formalizing mathematical proofs. However, it is often the case that a particular application of such types needs not only to inductively generate a collection of objects, but also to make identifications between the objects. In classical mathematics one can first generate and then identify, using the Axiom of Choice to lift infinitary constructions to the quotient. HITs can allow one to avoid such non-constructive uses of choice by inter-twining generation and identification. Perhaps more important than the constructive/non-constructive issue is that simultaneously declaring how to generate and how to identify can be a very natural way of defining some construct from the user's point of view. This is why HITs promise to be so useful, once we have robust and convenient mechanisms in theorem-proving systems for defining HITs and defining functions out of HITs. Although some HITs have been axiomatized in various systems, at the moment the only system I know of with built in support for defining quite general forms of HIT and using them is the implementation of cubical type theory [3] within recent versions of the Agda system [11].

The higher dimensional aspect of identity in cubical type theory is fascinating; nevertheless, the simpler one-dimensional version of identity, in which one has uniqueness of identity proofs (UIP), is adequate for many applications. Although some regard UIP as a bug of early versions of Agda with it's original form of dependent pattern matching [5], it is by choice a feature of the Lean prover [2]. Altenkirch and Kaposi [1] have termed the one-dimensional version of HITs *quotient inductive types* (QITs) and they promise to be very useful even in the setting of type theory with UIP.

In this talk I survey some of these developments, including a recent reduction of QITs to quotients [6], and the prospects for better support in theorem-provers for quotient constructions.

2012 ACM Subject Classification Theory of computation → Type theory; Theory of computation → Constructive mathematics

Keywords and phrases dependent type theory, quotient, quotient inductive type, theorem-proving systems

Digital Object Identifier 10.4230/LIPICs.FSCD.2020.3

Category Invited Talk

References

- 1 T. Altenkirch and A. Kaposi. Type theory in type theory using quotient inductive types. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '16, pages 18–29, New York, NY, USA, 2016. ACM.
- 2 J. Avigad, L. de Moura, and S. Kong. *Theorem Proving in Lean*, 3.4.0 edition, February 2020.
- 3 C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. Cubical type theory: A constructive interpretation of the univalence axiom. In T. Uustalu, editor, *21st International Conference on*



© Andrew M. Pitts;

licensed under Creative Commons License CC-BY

5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020).

Editor: Zena M. Ariola; Article No. 3; pp. 3:1–3:2

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- Types for Proofs and Programs (TYPES 2015)*, volume 69 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:34, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- 4 R. L. Constable, S. F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, D. J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, J. T. Sasaki, and S. F. Smith. *Implementing Mathematics with the Nuprl Proof Development System*. Prentice-Hall, Inc., USA, 1986.
 - 5 T. Coquand. Pattern matching with dependent types. In B. Nordström, K. Petersson, and G. D. Plotkin, editors, *Proceedings of the 1992 Workshop on Types for Proofs and Programs, Båstad, Sweden*, pages 66–79, June 1992.
 - 6 M. P. Fiore, A. M. Pitts, and S. C. Steenkamp. Constructing infinitary quotient-inductive types. In J. Goubault-Larrecq and B. König, editors, *23rd International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2020)*, volume 12077 of *Lecture Notes in Computer Science*, pages 257–276. Springer, 2020.
 - 7 M. Hofmann. *Extensional Concepts in Intensional Type Theory*. PhD thesis, University of Edinburgh, 1995.
 - 8 P. L. Lumsdaine and M. Shulman. Semantics of higher inductive types. *Mathematical Proceedings of the Cambridge Philosophical Society*, page 1–50, 2019.
 - 9 B. Nordström, K. Petersson, and J. M. Smith. *Programming in Martin-Löf’s Type Theory*. Oxford University Press, 1990.
 - 10 The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations for Mathematics*. <http://homotopytypetheory.org/book>, Institute for Advanced Study, 2013.
 - 11 A. Vezzosi, A. Mörtberg, and A. Abel. Cubical agda: A dependently typed programming language with univalence and higher inductive types. *Proc. ACM Program. Lang.*, 3(ICFP):87:1–87:29, July 2019.