An extensible equality checking algorithm for dependent type theories.

Zbl 07471706

Summary: We present a general and user-extensible equality checking algorithm that is applicable to a large class of type theories. The algorithm has a type-directed phase for applying extensionality rules and a normalization phase based on computation rules, where both kinds of rules are defined using the type-theoretic concept of object-invertible rules. We also give sufficient syntactic criteria for recognizing such rules, as well as a simple pattern-matching algorithm for applying them. A third component of the algorithm is a suitable notion of principal arguments, which determines a notion of normal form. By varying these, we obtain known notions, such as weak head-normal and strong normal forms. We prove that our algorithm is sound. We implemented it in the Andromeda 2 proof assistant, which supports user-definable type theories. The user need only provide the equality rules they wish to use, which the algorithm automatically classifies as computation or extensionality rules, and select appropriate principal arguments.

MSC:
03B70 Logic in computer science
68-XX Computer science

Keywords:
type theory; equality checking; proof assistant

Full Text: arXiv Link

References:
[7] Jesper Cockx and Andreas Abel. Sprinkles of extensionality for your vanilla type theory. In 22nd International Conference on Types for Proofs and Programs TYPES 2016, University of

Edited by FIZ Karlsruhe, the European Mathematical Society and the Heidelberg Academy of Sciences and Humanities
© 2022 FIZ Karlsruhe GmbH


This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.