

Abstract

This thesis investigates how to express and reason about relational concepts and methods inside the constructive logical framework of Martin-Löf's monomorphic type theory. We cover several areas where the notion of relation is central, and show how to formalize the basic concepts of each area. This formalization is carried out in a computer proof assistant for Martin-Löf's type theory called Agda/Alfa, and this software allows the practical handling of the numerous technical details involved in such a task.

The first area we investigate is relations and relational systems themselves. Many of the relational systems in use within computer science are based on algebras that extend Boolean algebra, and hence are unsuitable for a constructive framework. Instead we use the constructive algebra of relations provided by the categorical theory of allegories. First, we introduce the notion of an E-allegory, the appropriate abstract notion of an allegory formalizable in Martin-Löf type theory. Then we set up two concrete instances: the E-allegory of E-relations, and the E-allegory of finite decidable E-relations. We investigate further properties of the E-allegories and note that only the latter satisfies the axioms of a power allegory and a Boolean allegory.

The second area is relational program construction and derivation. Martin-Löf type theory provides its own approach to program construction via the identification of propositions and types, and our goal is to supplement this methodology with techniques from relational programming. Here we note that Martin-Löf type theory is based on a functional notion of program, and provides a logical setting where we can reason about the connections between functional and relational notions of programs. We show some links and differences with type-theoretic program construction and derivation. We then show how recursive relations can be dealt with in type theory using the concept of relational catamorphism.

The third area is the relational database model. This data model has similarities and connections with relational algebras and allegories, but presents new problems in the way typings are handled via schemes and attributes. We set up an increasingly complex series of formalizations of the relational model, with different decidability characteristics and with respect to the way in which the scheme/attribute typing issues are handled. The final formalization, which is both the most complex and the most satisfactory, is built on an abstract finite set layer, and can be seen as an abstraction for implementations of database managers.

Keywords: formalized mathematics, relational systems, category theory, programming logics, constructive type theory, logical frameworks, relational database model.

