Abstract

Brouwer's continuity principle that all functions from the Baire space to natural numbers are continuous is provably false in intuitionistic dependent type theory, with existence in the formulation of continuity expressed as a Σ-type via the Curry–Howard interpretation. However, with an intuitionistic notion of anonymous existence, defined as the propositional truncation of Σ, the principle becomes consistent and can be validated in Johnstone's topological topos. On the other hand, any of these two intuitionistic conceptions of existence give the same, consistent, notion of uniform continuity for functions from the Cantor space to natural numbers, again valid in the topological topos. But the treatment of the topological topos is non-constructive in several respects.

The object of the thesis is to give a (constructive and hence) computational interpretation of type theory validating the above uniform continuity principle, so that type-theoretic proofs with the principle as an assumption have computational content, and in particular closed terms of natural number type evaluate to numerals.

For this, we develop a variation of the topological topos. The site we work with is the monoid of uniformly continuous endomaps of the Cantor space $\mathbf{2}^{\mathbb{N}}$ equipped with a subcanonical topology consisting of certain countably many finite covering families, which is suitable for predicative, constructive reasoning. Our variation of the topological topos consists of sheaves on this site. Our concrete sheaves, like those in the topological topos, can be described as sets equipped with a suitable continuity structure, which we call C-spaces, and their natural transformations can be regarded as continuous maps. We mainly work with C-spaces in the thesis because they have sufficient structure to give a computational interpretation of the uniform-continuity principle. For instance, C-spaces form a (locally) cartesian closed category with a natural numbers object. Moreover, there is a fan functional in the category of C-spaces that continuously calculates (minimal) moduli of uniform continuity.

The C-spaces in our topos correspond to the limit spaces in the topological topos, in the sense that they are the concrete sheaves of the respective toposes in which they live. Similarly to the approach to the Kleene–Kreisel continuous functionals via limit spaces, we can also calculate the Kleene–Kreisel continuous functionals within the category of C-spaces, by starting from the discrete space of natural numbers and closed under products and exponentials. The C-spaces provide a classically equivalent substitute for the traditional manifestations of the Kleene–Kreisel spaces, which admits a constructive treatment of the uniform continuity principle mentioned above. Moreover, if we assume in our meta-language that all functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous, then we can show constructively that the full type hierarchy is equivalent to the Kleene–Kreisel continuous hierarchy within C-spaces.

Using the cartesian closed structure of C-spaces and the natural numbers object, we build a model of Gödel's system T, in which the uniform-continuity principle, formulated as a skolemized formula with the aid of an additional constant with the type of the fan functional, is validated. With the same interpretation of the term language of system T, we build a realizability semantics of higher-type Heyting arithmetic, with continuous maps of C-spaces as realizers, and use the fan functional again to realize a formula of the uniform-continuity principle. Moreover, we validate the Curry–Howard formulation of the uniform-continuity principle in the locally cartesian closed category of C-spaces.

The construction of C-spaces and the verification of the uniform continuity principle have been formalized in intensional Martin-Löf type theory in Agda notation, which is available at http://cj-xu.github.io/ContinuityType/.

Certain extensions of type theory are needed for the type-theoretic development due to the presence of proof relevance and the absence of function extensionality in Martin-Löf type theory. To avoid such extensions that may destroy the computational content of the development, we can make use of setoids, which produces a long formalization. However, by adjusting the model construction and postulating the double negation of function extensionality, we manage to achieve our main aim of extracting computational content from type-theoretic proofs that use the uniform-continuity principle, in a relatively clean way. In practice, we have used Agda to implement the extraction of computational content.