A Continuous Computational Interpretation of Type Theories

by

Chuangjie Xu

A thesis submitted to
The University of Birmingham
for the degree of
Doctor of Philosophy

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham
February 27, 2015

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 tedious 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.

Acknowledgements

I owe uncountably many thanks to a few people who provided me with their warm encouragement and kind support in every respect during my PhD study at Birmingham.

The first person that I would like to thank here is my supervisor, Martín Escardó. It is impossible for me to complete the work including writing this thesis without his patient guidance and effective supervision. I would never forget his understanding and consolation when I was overwhelmed by the message that my mother stayed in hospital due to her serious illness.

I appreciate the valuable feedback from the members of my thesis group, Achim Jung and Jon Rowe. In addition, I would like to acknowledge Achim, who was the supervisor in my undergraduate final-year project, for leading me in the area of research, particularly in theoretical computer science.

I have benefited from the help, such as feedback and suggestions to my papers and presentations, from the members of Theory of Computation Group in School of Computer Science at Birmingham, especially Achim Jung, Steve Vickers and Paul Levy.

During my funded stay in Institut Henri Poincaré, Paris, for Semantics of proofs and certified mathematics, I had interesting and beneficial discussions with Thierry Coquand, Peter Dybjer, Simon Huber, Guilhem Jaber, Bassel Mannaa, Per Martin-Löf, Paul-André Melliès, Anders Mörtberg, and Thomas Streicher.

I should also mention that my PhD study at Birmingham was financially supported by the University Graduate School with the Doctoral Elite Researcher Scholarship which made this work possible.

My family always unconditionally supports me and all my crucial decisions, such as taking the exchange project to complete my bachelor degree at Birmingham, and pursuing a doctorate degree.

Words cannot express my love and thanks to Min Tong, my sweet fiancee, for understanding my busy research life and for looking after me during the final, abustle phase of my writing up.

My special thanks also go to Nicolas, the little son of Martín, for saving me from the endless research, for his singular gift (a leaf from the plant in Martín's office), for his elusive abstract painting, and for asking me tough questions such as

"Chuangjie, what's your house number?" "Two-eleven." "Why?" "Ur..."

Contents

CHAPTER 1

Introduction

We investigate the compatibility of intensional Martin-Löf type theory (MLTT) [60, 61, 69] with Brouwerian continuity principles [7, 13, 73]. We consider the following two basic and canonical such principles:

(Cont) All functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are continuous.

(UC) All functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous.

Our main aim is to give a computational interpretation of type theory validating them, so that proofs in MLTT with (Cont) and (UC) as assumptions have computational content. However, instead of using computability theory to build such a model, we reason constructively, in Chapters 3–6, so that the computational content is implicit. We make it explicit in Chapters 7 and 8 by formalizing the model in MLTT itself, in Agda notation [11, 12, 63]. Because constructions and proofs in MLTT or Agda are programs in a literal sense, we can "run" our model directly to compute moduli of uniform continuity, as illustrated by some Agda experiments in Chapter 8.3.

The precise formulations of the above continuity principles are

(Cont) $\forall(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}).\ \forall(\alpha\colon \mathbb{N}^{\mathbb{N}}).\ \exists(m\colon \mathbb{N}).\ \forall(\beta\colon \mathbb{N}^{\mathbb{N}}).\ \alpha =_m \beta \Rightarrow f\alpha = f\beta,$

(UC) $\forall(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \exists(m\colon \mathbb{N}).\ \forall(\alpha, \beta\colon \mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \Rightarrow f\alpha = f\beta,$

where $\alpha =_m \beta$ means that the sequences α and β agree at the first m positions. Intuitively, the continuity principle (Cont) says that the value of the function f at the infinite sequence α depends only on a finite prefix of α, while the uniform-continuity principle (UC) strengthens this to say that the length of such a prefix does not depend on α. In classical mathematics, of course (Cont) implies (UC) using the compactness of the Cantor space $\mathbf{2}^{\mathbb{N}}$ [67]. The Baire space $\mathbb{N}^{\mathbb{N}}$ is not even locally compact, and, in fact, an example of a continuous but not uniformly continuous function $f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ is $f(\alpha) = \alpha(\alpha(0))$.

Johnstone's topological topos [48], among other toposes [38, 74], validates these principles. However, as illustrated in Chapter 3.1, the treatment of the topological topos is non-constructive in several respects. Our model, discussed below and defined in Chapter 3, is a variation of the topological topos, explicitly designed to allow a constructive treatment of (UC).

An important point is that, because we are considering the formulation of (Cont) and (UC) in MLTT, what we are interested in is their Curry–Howard (CH) interpretations

$$\text{(CH-Cont)}\quad \Pi(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}).\ \Pi(\alpha{:}\mathbb{N}^{\mathbb{N}}).\ \Sigma(m{:}\mathbb{N}).\ \Pi(\beta{:}\mathbb{N}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta,$$

$$\text{(CH-UC)}\quad \Pi(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \Sigma(m{:}\mathbb{N}).\ \Pi(\alpha,\beta{:}\mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta.$$

Perhaps surprisingly, (CH-Cont) is actually false in (intensional and hence in extensional) MLTT [31], before we consider any model. We recall the proof of this fact in Chapter 2.1. But the essence of the problem is that using (CH-Cont) and projections, we can define a modulus-of-continuity functional

$$M\colon (\mathbb{N}^{\mathbb{N}} \to \mathbb{N}) \to \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$$

such that

$$\Pi(\alpha,\beta{:}\mathbb{N}^{\mathbb{N}}).\ \alpha =_{Mf\alpha} \beta \to f\alpha = f\beta.$$

Whereas all functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ may be continuous, it is contradictory to assume that there is a functional M that finds moduli of continuity, because this can be used to define a non-continuous function $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$, as shown in Theorem 2.1.1. In particular, in the topological topos, all functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are continuous, but there is no continuous modulus-of-continuity functional M, where the continuity of M is taken in the sense of sequential spaces, which form a full subcategory of the topological topos, as explained below.

On the other hand, (CH-UC) is in some sense equivalent to (UC), as proved in Chapter 2.3. In order to understand this, and also the difference between (Cont) and (CH-Cont), we first consider toposes [49, 59]. Any topos, having a subobject classifier, can interpret the quantifiers $\forall$ and $\exists$, and, being locally cartesian closed, can interpret the type formers Π and Σ [66], and hence provides a model which simultaneously has (Cont) and (UC), as subobjects of the terminal object 1, and (CH-Cont) and (CH-UC), as certain objects. For example, in the topological topos, by the above discussion, (Cont) is the object 1, but (CH-Cont) is the initial object 0. Again (UC) is the object 1, but (CH-UC) is an object with a global point. In general, in any topos, we can show that (UC) is 1 if and only if (CH-UC) has a global point. Such a global point is a modulus-of-uniform-continuity functional, or a *fan functional*. Thus, in particular, in the topological topos, although it is not possible to continuously find moduli of continuity of functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$, it is possible to continuously find moduli of uniform continuity of functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$.

The discussion of the previous paragraph can be recast in type theory, as done in Chapter 2.2, if we extend it with a type former $\|-\|$ for so-called *bracket types* or *propositional truncations* [2, 72]. We interpret $\|X\|$ in a topos as the image of the unique map $X \to 1$. The idea is that $\|X\|$ is a type representing the truth-value of the assertion that the type X is inhabited, without revealing any inhabitant of X. Then (Cont) and (UC) can be equivalently formulated as

$$\text{(Cont)}\quad \Pi(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}).\ \Pi(\alpha{:}\mathbb{N}^{\mathbb{N}}).\ \|\ \Sigma(m{:}\mathbb{N}).\ \Pi(\beta{:}\mathbb{N}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta\ \|,$$

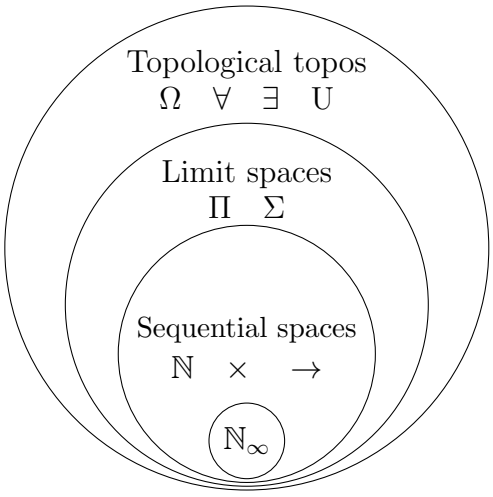
$$\text{(UC)}\quad \Pi(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \|\ \Sigma(m{:}\mathbb{N}).\ \Pi(\alpha,\beta{:}\mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta\ \|.$$

Given these type-theoretic formulations, we prove in Theorem 2.3.1, within type theory extended with propositional truncation, that (UC) is logically equivalent (CH-UC).

Our model validates (CH-UC), and hence (UC), reasoning constructively. Of course it cannot validate (CH-Cont) because it is provably false in MLTT. But, reasoning *non-constructively* in our meta-theory, it validates (Cont). The reason is that, using (UC), all

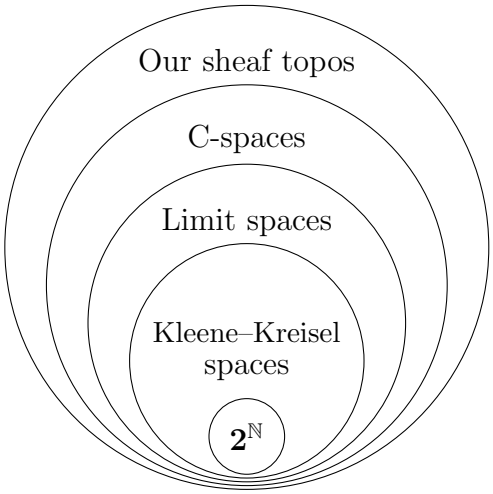
functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are (uniformly) continuous on compact subsets of $\mathbb{N}^{\mathbb{N}}$, and this gives their continuity, with a proof by contradiction, which amounts to saying that $\mathbb{N}^{\mathbb{N}}$ is a compactly generated space [33]. To avoid classical reasoning, we can instead work with the model considered by van der Hoeven and Moerdijk [74], which we discuss after we introduce our model.

To build the topological topos, one starts with the monoid of continuous endomaps of the one-point compactification $\mathbb{N}_\infty$ of the discrete natural numbers, and then takes sheaves for the canonical topology of this monoid considered as a category. Several convenient categories of spaces are fully embedded in the topological topos:



Firstly, the one-object full subcategory of topological spaces containing $\mathbb{N}_\infty$ (or the monoid for the definition of the topological topos) is of course fully embedded in the topos, by the Yoneda Lemma. The category of sequential spaces, which is cartesian closed and has a natural numbers object, and thus gives a model of Gödel's system T, is also fully embedded. But this category is not locally cartesian closed, and hence is not enough to model dependent types. The larger category of Kuratowski limit spaces, which also arises as the full subcategory of subobjects of sequential spaces in the topos (and hence are called subsequential spaces by Johnstone) is locally cartesian closed [62]. Moreover, limit spaces are precisely the *concrete sheaves* in the sense of [3]. The subobject classifier is not (isomorphic to) a limit space, and hence logic lives outside the realm of "spaces". Using Streicher's construction [70] for arbitrary Grothendieck toposes, assuming a Grothendieck universe in set theory, one can build an object U that serves as a Martin-Löf universe. But, again this object is not a limit space [34].

We work with a different site for our variation of the topological topos. Instead of the monoid of continuous endomaps of $\mathbb{N}_\infty$, we take the monoid C of uniformly continuous endomaps of the Cantor space $\mathbf{2}^{\mathbb{N}}$. Rather than working with the canonical topology, we consider a subcanonical one, consisting of certain countably many finite covering families, which is suitable for predicative, constructive reasoning. We call it the *uniform-continuity* coverage, because the coverage axiom specialized to our situation amounts to the fact that the elements of the monoid C are the uniformly continuous functions (Chapter 3.2.1). Our model consists of sheaves on this uniform-continuity site. Similarly to the topological topos, certain spaces reside in our sheaf topos, as indicated in the following diagram:



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. The idea is that we "topologize" a set X by choosing a designated collection of maps $\mathbf{2}^{\mathbb{N}} \to X$, called *probes*, that we want, and hence declare, to be continuous. We call this collection the C-*topology* on X, and say that X is a C-*space*. As discussed in Chapter 3.3.1, our C-spaces can be regarded as a variation of Spanier's quasi-topological spaces [68]. In this thesis, we mainly work with C-spaces because:

1. C-spaces admit a more concrete and intuitive description.

2. C-spaces have sufficient structure, as discussed below, for our aim of giving a computational interpretation of the uniform-continuity principle (CH-UC).

3. C-spaces are easier to work with, compared to sheaves, regarding the type-theoretic implementation, as discussed in Chapter 7.3.

One disadvantage of C-spaces is their lack of a subobject classifier and universe (or object classifier). Similarly to the situation in the topological topos, the subobject classifier in our sheaf topos is not a C-space. Hence, C-spaces are insufficient to interpret the quantifiers $\forall$ and $\exists$, and thus cannot model the principles (Cont) or (UC). Moreover, the universe object in our sheaf topos built using Streicher's construction again is not a C-space either. Thus, working with C-spaces, we cannot model Martin-Löf's universes. We demonstrate how our sheaves form a model of dependent types in Chapter 6.5, in the sense of a category with families [29]. We also attempt to interpret the universe, following Coquand's construction of presheaf models [21], but we have more questions than answers about the interpretation of the universe in our model, which we leave as an important open problem (Chapter 9.6).

Similarly to limit spaces, our C-spaces also form a (locally) cartesian closed category as shown in Chapter 3.3.2. The constructions are the same as those in the category of sets, with suitable C-topologies. For instance, to get products of C-spaces we "C-topologize" the cartesian products, and to get exponentials of C-spaces we "C-topologize" the sets of continuous maps. As proved in Chapter 3.3.3, the category of C-spaces has a natural numbers object $\mathbb{N}$, which is a discrete C-space, *i.e.* any function from it is continuous.

Specifically, the probes on $\mathbb{N}$ are precisely the uniformly continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$. The Yoneda Lemma, specialized to our topos, says that a map $\mathbf{2}^{\mathbb{N}} \to X$ is continuous (in the sense of C-spaces) if and only if it is a probe on X. In the special case $X = \mathbb{N}$, it follows that all morphisms $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ amount to uniformly continuous functions. Using this fact, in Chapter 3.5, we construct a *fan functional* in the category of C-spaces that continuously calculates minimal moduli of uniform continuity of maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$. This functional is used to validate the uniform-continuity principle in type theory, as discussed below.

From the previous discussion, we see that 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. We also show that limit spaces can be fully embedded in the category of C-spaces, as proved in Chapter 4.2. An important fact is that any topological space X becomes a limit space, by equipping it with all topologically continuous maps $\mathbb{N}_\infty \to X$, and becomes a C-space, by equipping it with all topologically continuous maps $\mathbf{2}^{\mathbb{N}} \to X$. Thus both objects $\mathbb{N}_\infty$ and $\mathbf{2}^{\mathbb{N}}$ of our topos and of the topological topos are limit spaces and C-spaces. Given a limit space X, if we take all continuous maps $\mathbf{2}^{\mathbb{N}} \to X$ (in the sense of limit spaces) to be the probes, then X becomes a C-space. This gives the full embedding of limit spaces into C-spaces. Given a C-space X, the continuous maps $\mathbb{N}_\infty \to X$ (in the sense of C-spaces) form a limit-structure on X. This gives a left adjoint to the embedding, which preserves finite products. Therefore, limit spaces form an exponential ideal of the category of C-spaces.

One of the well known approaches to the Kleene–Kreisel continuous functionals [64, 65, 56, 57] is to work with limit spaces: we start from the discrete space of natural numbers, close under products and exponentials, and then obtain a full subcategory which is equivalent to any of the known formulations of Kleene–Kreisel spaces, as briefly recalled in Chapter 4.1. When restricted to the objects in this subcategory, the embedding mentioned above becomes an equivalence (in fact, even an isomorphism); therefore, the Kleene–Kreisel spaces can be calculated within C-spaces by starting from the discrete space of natural numbers and iterating products and exponentials. We emphasize that the proof of this fact, in Chapter 4.2, is non-constructive. For example, the proof that the natural numbers objects in the two categories coincide uses an argument by contradiction. But we also emphasize that Chapter 4.2 is the only part of our work that contains non-constructive arguments. Since our development of C-spaces, including the cartesian closed structure (Chapter 3.3.2) and the fan functional (Chapter 3.5), is constructive, our 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 discussed above.

Kleene–Kreisel spaces form a well-known simple-type hierarchy in which all functionals are continuous. The full type hierarchy is the smallest full subcategory of sets containing the natural numbers and closed under products and exponentials. One interesting observation in Chapter 4.3 is that, when assuming the Brouwerian axiom that all set-theoretic functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous, the full type hierarchy is equivalent to the Kleene–Kreisel continuous hierarchy within C-spaces, which is reminiscent of Fourman's recent work [39] on reflection. The proof is constructive, and it is interesting that other Brouwerian axioms such as more general forms of continuity or Bar Induction are not needed to prove the equivalence.

Using the cartesian closed structure of C-spaces and its natural numbers object, in Chapter 5.1 we build a model of system T. We firstly recover a well known result, namely that any T-definable function $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ in the full type hierarchy is uniformly continuous, by establishing a logical relation between set-theoretic interpretation (*i.e.* the full type hierarchy) and the C-space interpretation (*i.e.* the Kleene–Kreisel continuous hierarchy) of simple types. Since system T is a quantifier-free theory, in order to formulate the uniform-continuity principle, we add a constant with the type of the fan functional, and then skolemize the formula (UC) with the aid of this additional constant. Using the definition of the fan functional, we easily show that (the skolemization of) the principle (UC) is validated, by interpreting the constant as the fan functional. Then in Chapter 5.2, with the same interpretation of the term language of T, we build a realizability semantics of the higher-type Heyting arithmetic HA^{ω} [7, 73], with continuous maps of C-spaces as realizers, and use the fan functional again to realize a formula of (UC). Although, in HA^{ω}, the uniform-continuity principle is formulated with quantifiers $\forall$ and $\exists$, they are realized by function spaces and binary products in the category of C-spaces, instead of using the local cartesian closed structure.

Following Seely's interpretation of (extensional) Martin-Löf type theory in locally cartesian closed categories [66], we validate the uniform-continuity principle (CH-UC) in the locally cartesian closed category of C-spaces in Chapter 6.2. The proof to this result is essentially the same as the ones given to System T and HA^{ω}, as both theories can be regarded as subsystems of MLTT. In order to address a well-known coherence issue in Seely's interpretation [25], regarding the interpretation of substitution as pullback, we work with categories with families (CwFs) [29] in Chapters 6.4 and 6.5. We show how to give CwF structures to the categories of C-spaces and of sheaves to get models of dependent types.

The above work, presented in Chapters 3–6, is developed within an informal constructive meta-theory, similar to Bishop's mathematics in style [10]. Then in Chapter 7 we develop it in a formal meta-theory, in Martin-Löf's style [61]. We emphasize that our model (in both its informal and formal manifestations) is developed in a minimalistic constructive meta-theory, which is compatible with classical mathematics. In particular, no constructively contentious principles, such as continuity axioms, fan theorem and Bar induction [7, 73], or impredicativity are assumed in the meta-language.

Because Martin-Löf type theory is "proof-relevant" [61], the development in Chapter 7 needs to perform a number of adjustments, or refinements, to the informal development of Chapters 3–6. For instance, the collection of uniformly continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ is formulated as a Σ-type, *i.e.* a uniformly continuous map is a pair consisting of a underlying map $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ and a witness of uniform continuity. When formalizing the proof that the domain $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$ of the fan functional is a discrete space (Lemma 3.5.1), if we attempt to prove an equality of two uniformly continuous maps, *i.e.* two pairs, we would be able to only obtain an equality of their underlying maps, which is not sufficient, because even for the same map there could be many different witnesses of uniform continuity. By requiring the existence of a *minimal* modulus of uniform continuity, the type that expresses that a map is uniformly continuous can have at most one inhabitant, and with this refinement we can complete the formalization of the discreteness proof of the space $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$.

Another difficulty of the type-theoretic formalization, as discussed in Chapter 7.1, is caused by the lack of *function extensionality* (funext) in MLTT. Certain issues with this

arise in the developments of: (1) exponentials of C-spaces, (2) discrete C-topologies, and (3) the fan functional. To handle these issues, we developed mainly four approaches, among others, all of them implemented in Agda and available at [76]:

1. *Use setoids.* This well known approach [41], which is also at the heart of Bishop's approach to constructive analysis [10], consisting of the use of types equipped with equivalence relations, works here with no surprises. But the drawback, as usual, is that it gives a long formalization that obscures the essential aspects of the constructions and proofs.

2. *Simply postulate (funext).* This is of course the easiest approach, but would potentially destroy the computational content of formal proofs, because then (funext) becomes a constant without a computational rule. Thus, although we obtain a clean formalization, we potentially lose computational content, which would defeat the main aim of the thesis.

3. *Postulate (funext) within a computationally irrelevant field.* After the previous approach (2) was completed, we observed that our uses of (funext) do not really have computational content, and we used a feature of Agda, called irrelevant fields [1], to formulate and prove this observation. In practice, we actually needed to slightly modify approach (2) to make this idea work, as discussed in Chapter 7.1. In any case, the drawback is that it requires the extension of type theory with such irrelevant fields, which we would prefer to avoid.

4. *Postulate the double negation of (funext).* In turn, after we completed approach (3), we observed that it does not really depend on the nature of irrelevant fields, but only on the fact that irrelevant fields form a monad T with $T\emptyset \to \emptyset$. As is well known, double negation is the final such monad. There are two advantages with this approach: (i) we do not need to work with a non-standard extension of MLTT, and (ii) postulating negative, consistent axioms does not destroy computational content [22].

With the last approach (4), we achieve our main aim of extracting computational content from proofs that use the axiom (CH-UC), in a relatively clean way, avoiding the usual bureaucracy associated to setoids.

1.1 Summary of contributions

The main contributions of this work mentioned above are summarized as follows:

1. The uniform-continuity principle (UC) is *logically equivalent* to its Curry–Howard interpretation (CH-UC) (Chapter 2.3).

2. Without assuming Brouwerian axioms, we show constructively that the category of C-spaces has a *fan functional* $(\mathbf{2}^{\mathbb{N}} \to \mathbb{N}) \to \mathbb{N}$ that continuously calculates moduli of uniform continuity of maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ (Chapter 3.5).

3. Kleene–Kreisel continuous functionals can be calculated within the category of C-spaces (Chapter 4.2).

The proof here is non-constructive (as are the proofs for the traditional approaches). But we claim that C-spaces provide a good substitute of the traditional approaches to the Kleene–Kreisel spaces for the purposes of constructive reasoning about continuity principles.

4. If we assume the Brouwerian principle that all set-theoretic 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 (Chapter 4.3).

5. C-spaces give a model of system T with a uniform-continuity principle, expressed as a skolemization with the aid of a fan-functional constant (Chapter 5.1).

6. C-spaces give a realizability semantics of HA^{ω}, in which the uniform-continuity principle is realized by the fan functional (Chapter 5.2).

7. C-paces give a model of dependent types with a uniform-continuity principle, expressed as a closed type via the Curry–Howard interpretation (Chapter 6.2).

8. We give a constructive treatment of C-spaces (Chapters 3.3–3.5) suitable for development in a predicative intuitionistic type theory in the style of Martin-Löf [60, 61], which we formalized in Agda notation [11, 12, 63] for concrete computational purposes, and whose essential aspects are discussed in Chapters 7 and 8.

Among the above, (1) was merged with [31] to produce the paper [36] which has been submitted for publication; (2), (5), (6), and part of (8) were presented in our TLCA paper [77]; and (3), (4), (7), and part of (8), appeared as new results in its full version [35], submitted for publication. All these results have been refined and expanded upon in this thesis.

1.2 Summary of related work

We have already mentioned connections with earlier work in the above discussion. In this section we briefly summarize this. Our contributions build upon the following work:

1. Johnstone's paper *On a topological topos* (1979) [48],

2. Fourman's papers *Notions of choice sequence* (1982) [37], *Continuous truth I* (1984) [38], and *Continuous truth II* (2013) [39],

3. van der Hoeven and Moerdijk's paper *Sheaf models for choice sequences* (1984) [74],

4. Spanier's paper *Quasi-topologies* (1963) [68],

5. Bauer and Simpson's unpublished work *Continuity begets continuity* (2006) [6], and

6. Coquand and Jaber's paper *A note on forcing and type theory* (2010) [23] and *A computational interpretation of forcing in type theory* (2012) [24].

Johnstone, Fourman, van der Hoeven and Moerdijk, among the above, work with sheaf toposes on different sites, and so do we as discussed above and in Chapter 3.2. The concrete sheaves in these toposes can be regarded as variations of Spanier's quasi-topological spaces, and so can ours as discussed in Chapter 3.3.1.

As mentioned earlier, to build the topological topos, one starts with the monoid of continuous endomaps of the one-point compactification of the discrete natural numbers, and then takes sheaves for the canonical topology of this monoid considered as a category. Working non-constructively, one can show that the continuity principles (Cont) and (UC) are validated by the topological topos.

Fourman works with a site whose underlying category is the semilattice of finite sequences of natural numbers under the prefix order. He shows that general principles of continuity, local choice and local compactness hold for his models. With an analysis of iterated categorical extensions, he concludes a principle of predicative reflection. Our result that (UC) implies the equivalence of the full type hierarchy and the Kleene–Kreisel continuous hierarchy (Chapter 4.3) is analogous to his reflection principle.

Inspired by Fourman's work, van der Hoeven and Moerdijk consider the monoid of continuous endomaps of the Baire space $\mathbb{N}^{\mathbb{N}}$ and a so called *open cover topology* to get a sheaf model of countable choice, continuity principle for maps $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$, and Bar induction. To be precise, the continuity principle validated by their model is its logical formulation (Cont).

Bauer and Simpson's work can be seen as taking place in the topological topos. They consider a weaker notion of limit space, and develop a model of (some form of) predicative constructive mathematics, in which every function of complete (separable) metric spaces is sequentially continuous. They claim that their argument is predicative and constructive.

Our work is also related to Coquand and Jaber's forcing model, which instead uses the semilattice of finite binary sequences under the prefix order as the underlying category of the site, modelling the idea of a generic infinite binary sequence. They iterate their construction in order to be able to model the fan functional, and our model can be regarded as accomplishing this iteration directly in a single step (personal communication with Coquand). A difference is that their approach is syntactical rather than semantical: instead of constructing a model, they decorate the operational semantics of type theory with forcing information.

1.3 Prerequisites

The prerequisites for reading the thesis include:

1. Topology [67, 71]: topological space, continuous function, compact Hausdorff space, Cantor space $\mathbf{2}^{\mathbb{N}}$, Baire space $\mathbb{N}^{\mathbb{N}}$, one-point compactification $\mathbb{N}_\infty$ of natural numbers.

2. Category theory [58, 49, 59]: category, functor, natural transformation, limit, adjunction, cartesian closed category, locally cartesian closed category, natural numbers object, subobject classifier, isomorphism of categories, equivalence of categories.

3. Topos theory [49, 59]: site, Grothendieck topology, canonical topology, subcanonical topology, presheaf, sheaf.

Some basic familiarity with recursion theory [64], *e.g.* the Kleene–Kreisel continuous functionals [64, 65, 56, 57], would be helpful when reading Chapter 4, where the first

section briefly presents some necessary background, and recalls the approach to Kleene–Kreisel continuous functionals via limit spaces, to make the chapter self-contained.

Some knowledge of constructive mathematics (*e.g.* the first three chapters of [7] and Chapter 4 of [73]), including that of constructive type theory (*e.g.* [60, 61, 69]), would be necessary to understand Chapter 2, regarding the type-theoretic formulations of continuity principles, and Chapters 7 and 8, regarding our type-theoretic development and Agda implementation. But the readers unfamiliar with constructive mathematics would not be in disadvantage except in these chapters.

1.4 Organization

The remaining chapters of the thesis are organized as follows.

Chapter 2 investigates the Curry–Howard formulations of the two fundamental continuity principles, (Cont) and (UC). The former is provably false in intensional Martin-Löf type theory and thus fails in any topos. The latter, which is the one that we are working with in this thesis, is logically equivalent to the logical formulation.

Chapter 3 develops a variation of the topological topos, consisting of sheaves on a certain uniform-continuity site. In particular, C-spaces, corresponding to concrete sheaves, 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 moduli of uniform continuity of maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$.

Chapter 4 shows how the Kleene–Kreisel continuous functionals can be calculated within C-spaces. When assuming that all set-theoretic functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous, the full type hierarchy is equivalent to the Kleene–Kreisel continuous hierarchy within C-spaces.

Chapter 5 employs C-spaces to model Gödel's system T with a skolemization of (UC), and to realizes (UC) in the intuitionistic arithmetic HA^{ω} of finite types, with the aid of the fan functional.

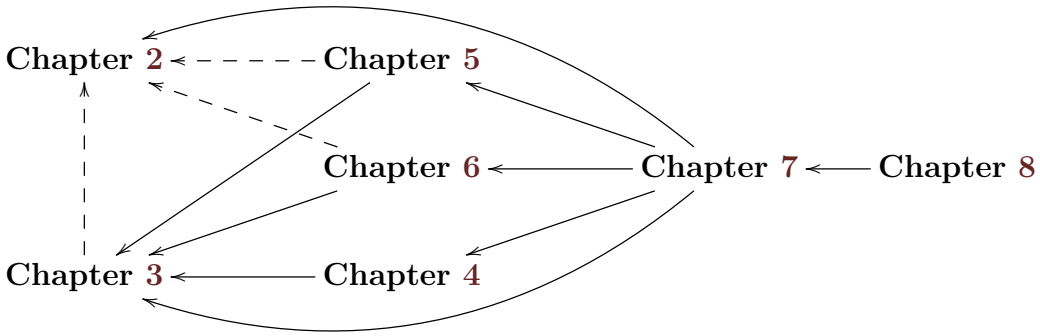
Chapter 6 validates the Curry–Howard interpretation of (UC) in the locally cartesian closed category of C-spaces using the fan functional, and demonstrates how C-spaces and sheaves form models of dependent type via the notion of category with families.

Chapter 7 discusses the main difficulties of developing the model in type theory, provides a few approaches to overcome them, and explores the feasibility of internalizing models of type theory.

Chapter 8 presents some examples of our Agda implementation to demonstrate how to "run" the model to compute moduli of uniform continuity.

Chapter 9 concludes with a summary of main results in the thesis, as well as a few interesting directions of further research related to the thesis.

The interdependence of the main chapters (2–8) in this thesis is given by the following diagram, where the arrow $B \longrightarrow A$ represents the dependence of B on A, and the dashed one $B \dashrightarrow A$ means that A is helpful but not necessarily needed to understand B.



The purpose of presenting the material in Chapter 2 is to clarify the aim of this thesis, which is to extract computational content from type-theoretic proofs that use the principle (CH-UC) as an assumption. Thus reading Chapter 2 would be helpful for readers to understand how our model is constructed (Chapter 3) and which formulation of the uniform-continuity principle is validated (Chapters 5 and 6).

Chapter 4 mainly demonstrates how the Kleene–Kreisel continuous functionals are calculated within our model which is developed in Chapter 3, and hence is independent of the other main chapters, except that the constructive proof of the hierarchy equivalence is briefly discussed in Chapter 7.

Chapters 5 and 6 employ C-spaces developed in Chapter 3 to model simple and dependent type theories, and then validate the uniform-continuity principle. The work presented in these three chapters, as highlighted above, are developed within an informal constructive meta-language. Thus unfamiliarity with constructive mathematics would not be a serious problem when reading them.

Chapters 7 and 8 investigate the constructive aspects of the work presented in the precedent chapters.

CHAPTER 2

The formulation of continuity principles in type theory

Two of the simplest continuity principles considered in constructive mathematics are

1. (Cont) All functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are continuous.
2. (UC) All functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous.

For more general continuity principles, e.g. for metric spaces, and their relationship to other principles such as Bar Induction, the Fan Theorem, etc. see Troelstra and van Dalen [73] and Beeson [7].

The aim of this thesis is to develop a model of dependent type theory that validates (UC) directly. Moreover, we wish to develop such a model in a minimalistic constructive meta-theory, compatible with classical mathematics, even type theory itself. Although we will not model (Cont), in this chapter we discuss it in parallel with (UC) in order to illustrate certain phenomena that arises from the Curry-Howard interpretation of logic usually adopted in Martin-Löf type theory, and in order to clarify the main aim of the thesis, which is to validate the Curry–Howard interpretation of (UC).

The precise logical formulations of the above principles are

$$\text{(Cont)}\ \ \forall(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}).\ \forall(\alpha\colon \mathbb{N}^{\mathbb{N}}).\ \exists(m\colon \mathbb{N}).\ \forall(\beta\colon \mathbb{N}^{\mathbb{N}}).\ \alpha =_m \beta \Rightarrow f\alpha = f\beta,$$

$$\text{(UC)}\quad \forall(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \exists(m\colon \mathbb{N}).\ \forall(\alpha,\beta\colon \mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \Rightarrow f\alpha = f\beta,$$

where $\alpha =_m \beta$ means that the sequences α and β agree at the first m positions. Both principles are validated by Johnstone's topological topos [48], which is recalled in Section 3.1, among other toposes, such as [38, 74]. Under the Curry-Howard interpretation of logic, these principles become the types

$$\text{(CH-Cont)}\ \ \Pi(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}).\ \Pi(\alpha{:}\mathbb{N}^{\mathbb{N}}).\ \Sigma(m{:}\mathbb{N}).\ \Pi(\beta{:}\mathbb{N}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta,$$

$$\text{(CH-UC)}\quad \Pi(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \Sigma(m{:}\mathbb{N}).\ \Pi(\alpha,\beta{:}\mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta,$$

where the equality sign stands for the identity type (rather than judgemental equality). Perhaps surprisingly, (CH-Cont) can be proved to fail in intensional (and hence in extensional) MLTT [31, 36] (Section 2.1). Moreover, (CH-Cont) can also be considered in any

topos via its local cartesian closed structure, which allows one to interpret Π ad Σ, and always defines an initial object.

In contrast, we show that the principle (CH-UC) is not only consistent, but also equivalent to (UC) in a precise sense (Section 2.3). This shows that the topological topos validates (CH-UC), at least within a non-constructive meta-theory. The topological topos is briefly discussed in Section 3.1, within Chapter 3, which discusses a variation of the topological topos which allows us to model (CH-UC), and, moreover, can be developed within a minimal constructive meta-theory.

2.1 The Curry–Howard interpretation of (Cont)

The following theorem of intensional Martin-Löf type theory is due to Escardó [31, 36], with an argument that goes back to Kreisel [54]. We reason informally, but rigorously, in MLTT, where, as above, we use the equality sign to denote identity types. A formal proof, written in Agda notation, is provided in *loc. cit.*

Theorem 2.1.1. *If* $\Pi(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}).\Pi(\alpha\colon \mathbb{N}^{\mathbb{N}}).\Sigma(m\colon \mathbb{N}).\Pi(\beta\colon \mathbb{N}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta$ *then* $0 = 1$.

Proof. Let 0^ω denote the infinite sequence of zeros, that is, $\lambda i.0$, and let $0^n k^\omega$ denote the sequence of n many zeros followed by infinitely many k's. Then

$$(0^n k^\omega) =_n 0^\omega \quad \text{and} \quad (0^n k^\omega)(n) = k.$$

Assume $\Pi(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}).\Pi(\alpha\colon \mathbb{N}^{\mathbb{N}}).\Sigma(n\colon \mathbb{N}).\Pi(\beta\colon \mathbb{N}^{\mathbb{N}}).\alpha =_n \beta \to f(\alpha) = f(\beta)$. By projection, with $\alpha = 0^\omega$, this gives a modulus-of-continuity function

$$M\colon (\mathbb{N}^{\mathbb{N}} \to \mathbb{N}) \to \mathbb{N}$$

such that

$$\Pi(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}).\Pi(\beta\colon \mathbb{N}^{\mathbb{N}}).0^\omega =_{Mf} \beta \to f(0^\omega) = f(\beta). \tag{2.1}$$

We use M to define a non-continuous function $f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ and hence get a contradiction. Let

$$m :\equiv M(\lambda\alpha.0),$$

and define $f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ by

$$f(\beta) :\equiv M(\lambda\alpha.\beta(\alpha m)).$$

Observe that, by simply expanding the definitions,

$$f(0^\omega) = m.$$

By the defining property (2.1) of M,

$$\Pi(\beta\colon \mathbb{N}^{\mathbb{N}}).0^\omega =_{Mf} \beta \to m = f\beta. \tag{2.2}$$

Now we consider the two cases $Mf = 0$ and $Mf > 0$.

Assume $Mf = 0$. By (2.2),

$$\Pi(\beta\colon \mathbb{N}^{\mathbb{N}}).m = f\beta.$$

The choice $\beta i :\equiv i$ gives

$$m = f(\lambda i.i) = M(\lambda\alpha.\alpha m).$$

By the defining property (2.1) of M, this means that

$$\Pi(\alpha\colon \mathbb{N}^{\mathbb{N}}).0^\omega =_m \alpha \to 0 = \alpha m.$$

But this gives $0 = 1$ if we choose e.g. the sequence $\alpha = 0^m 1^\omega$.

Now assume $Mf > 0$ instead. For any $\beta\colon \mathbb{N}^{\mathbb{N}}$, by the continuity of $\lambda\alpha.\beta(\alpha m)$, by the definition of f, and by the defining property (2.1) of M, we have that

$$\Pi(\alpha\colon \mathbb{N}^{\mathbb{N}}).0^\omega =_{f\beta} \alpha \to \beta 0 = \beta(\alpha m).$$

Considering $\beta :\equiv 0^{Mf}1^\omega$, this gives

$$\Pi(\alpha\colon \mathbb{N}^{\mathbb{N}}).0^\omega =_m \alpha \to \beta 0 = \beta(\alpha m),$$

because $f\beta = m$ as $0^\omega =_{Mf} \beta$ and $f(0^\omega) = m$. Considering $\alpha = 0^m(Mf)^\omega$, this in turn gives $0 = \beta 0 = \beta(\alpha m) = \beta(Mf) = 1$. □

Because any topos interprets (extensional and hence intensional) MLTT, the above holds in particular in the topological topos, despite the fact that this topos validates the usual logical formulation of (Cont) with the logical quantifiers $\forall, \exists$ rather than the type formers Π, Σ. This apparent contradiction is discussed in the following section.

2.2 Relationship between $\exists, \forall$ and Σ, Π in a topos

In order to understand how it can be that (Cont) is valid in some models of some varieties of constructive mathematics [7, 73, 13], but absurd in intensional Martin–Löf type theory, we consider toposes, as they simultaneously have the quantifiers $\exists, \forall$ and the object formers Σ, Π, which have markedly different meanings. Those of $\exists, \forall$ come via the subobject classifier Ω, and those of Σ, Π come via the local cartesian closed structure of the topos [49].

The interpretation of a truth value in a topos is a sub-terminal object. For any object X of a topos, its support, written $\|X\|$, is the image of the unique map $X \to 1$. This expresses the inhabitedness of X as a truth value. Any map $P\colon X \to \Omega$ to the object of truth values can be regarded as a property of elements of X, and so we can form the truth value $\exists(x{:}X).P(x)$, or can be regarded as a family of sub-terminal objects indexed by X, and so we can form the object $\Sigma(x{:}X).P(x)$, and we have

$$\exists(x{:}X).P(x) \;\cong\; \|\Sigma(x{:}X).P(x)\|.$$

We also have

$$\forall(x{:}X).P(x) \;\cong\; \Pi(x{:}X).P(x)$$

because a product of sub-terminal objects is always sub-terminal. Thus, although $\exists, \forall$ and Σ, Π are related, they are different in general, as illustrated by (Cont), which is valid in the topological topos, and its Curry–Howard interpretation (CH-Cont), which is not.

However, sometimes there is no essential difference between $\exists$ and Σ. We show that this is the case for (UC) in Section 2.3. By the above discussion, the logical formulation

of (UC) can be equivalently expressed type-theoretically as

$$\Pi(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \|\Sigma(m\colon \mathbb{N}).\ \Pi(\alpha,\beta\colon \mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta\|.$$

This type is not isomorphic to the "untruncated" type

$$\Pi(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \Sigma(m\colon \mathbb{N}).\ \Pi(\alpha,\beta\colon \mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta$$

as there are in general many moduli $m\colon \mathbb{N}$ of uniform continuity of the same $f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$. But we prove that they are "logically equivalent", where we say that two objects X and Y are logically equivalent if we have maps $X \to Y$ and $Y \to X$.

We formulate and prove this in MLTT extended with a type constructor $\|-\|$. This extension can be interpreted in any topos as above, with identity types modelled by equalizers. This is a fragment of the type theory considered in the HoTT book [72], leaving out the univalence principle and higher inductive types. To formulate the support operation $\|-\|$, referred to as -1-truncation or *propositional truncation* in [72], we first define

$$\mathrm{isProp}\, A :\equiv \Pi(a,b\colon A).a = b.$$

for any type A. This says that a *proposition* is a type with at most one element. We have the introduction rule (or constructor)

$$|-|\colon X \to \|X\|$$

with the postulate

$$\mathrm{isProp}\, \|X\|$$

for any type X. The elimination rule (or recursion principle) is that for any proposition P and map $f\colon X \to P$ there is a map $\bar{f}\colon \|X\| \to P$ such that $\bar{f}|x| = f(x)$ holds judgementally for all $x\colon X$. Intuitively, the above specification of $\|X\|$ says it the quotient of X by the equivalence relation that identifies all elements of X (and this is the case in the topos interpretation).

From the existence of the truncation of the coproduct type $\mathbf{1} + \mathbf{1}$ (of two copies of the singleton type) and the judgemental equality obtained via the elimination rule, one can prove function extensionality [52], that is, for any type X and any type family $x\colon X \vdash Y(x)$,

$$\Pi(f,g\colon \Pi(x\colon X).Y(x)).\ (\Pi(x\colon X).fx = gx) \to f = g.$$

We emphasize that the proof of the logical equivalence of the two formulations of the uniform-continuity principle (Theorem 2.3.1) does use function extensionality, which is available in this extended MLTT.

Remark. In our Agda formalization [76], we in fact postulate a weaker form of propositional truncation than that of the HoTT book, due to the lack of the judgemental equality in the elimination rule. This does not seem to allow one to derive function extensionality, and hence we have to additionally postulate function extensionality in order to complete the formalization of the proof of Theorem 2.3.1.

2.3 The Curry–Howard interpretation of (UC)

Again, as in Section 2.1, we reason informally, but rigorously in type theory.

Theorem 2.3.1. *In intensional Martin-Löf type theory with propositional truncation, the proposition*

$$\Pi(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \|\Sigma(m\colon \mathbb{N}).\ \Pi(\alpha,\beta\colon \mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta\|$$

is logically equivalent to the type

$$\Pi(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \Sigma(m\colon \mathbb{N}).\ \Pi(\alpha,\beta\colon \mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta.$$

We use the following two lemmas from [51].

Lemma 2.3.2. *If X is any type and $f\colon X \to X$ is a constant function in the sense that $fx = fy$ for all $x, y\colon X$, then the type* $\mathrm{fix}(f) :\equiv \Sigma(x\colon X).x = fx$ *of fixed points of f is a proposition.*

This has a non-trivial proof and we refer the reader to *loc. cit.* But we include the proof of the second lemma:

Lemma 2.3.3. *For any type X we have $\|X\| \to X$ if X has a constant endomap.*

Proof. Let $f\colon X \to X$ be constant. Then the type $\mathrm{fix}(f)$ is a proposition by the previous lemma. Using the constancy witness of f, we can define a map $X \to \mathrm{fix}(f)$. According to the elimination rule of $\|-\|$ we get a map $\|X\| \to \mathrm{fix}(f)$. Then the composition with the first projection gives the desired result. □

We use this to prove the following:

Lemma 2.3.4. *If A is a family of types indexed by natural numbers such that*

1. *$A(n)$ is a proposition for any $n\colon \mathbb{N}$, and*
2. *$A(n)$ implies that $A(m)$ decidable for every $m < n$,*

then

$$\|\Sigma(n{:}\mathbb{N}).A(n)\| \to \Sigma(n{:}\mathbb{N}).A(n).$$

Proof. Given a pair $(n, a_n)\colon \Sigma(n{:}\mathbb{N}).A(n)$, we know that $A(m)$ is decidable for all $m < n$ and thus can find the minimal m such that $A(m)$, by search bounded by n, which gives a map $\mu\colon \Sigma(n{:}\mathbb{N}).A(n) \to \Sigma(n{:}\mathbb{N}).A(n)$. The witness of $A(m)$ is obtained from the witness of $A(m) + \neg A(m)$. Thus, for different $w, w'\colon \Sigma(n{:}\mathbb{N}).A(n)$, the map μ gives the same minimal m, i.e. $\mathrm{pr}_1(\mu(w)) = \mathrm{pr}_1(\mu(w')) = m$, but the witnesses of $A(m)$ can be different, because the witness of $A(m) + \neg A(m)$ depends on the input pair. Since $A(m)$ is is a proposition, the two witnesses $\mathrm{pr}_2(\mu(w))$ and $\mathrm{pr}_2(\mu(w'))$ of $A(m)$ are equal. Hence, the map μ is constant. Then Lemma 2.3.3 gives the desired result. □

The above three lemmas do not need function extensionality, but the application of the last lemma in the following argument does. Specifically, function extensionality is used to prove that the product type $\Pi(\alpha,\beta\colon \mathbf{2}^{\mathbb{N}}).\ \alpha =_n \beta \to f\alpha = f\beta$ is a proposition for any $f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ and $n\colon \mathbb{N}$.

Proof of Theorem 2.3.1. One direction is an immediate consequence of the fact that $X \to \|X\|$ for any type X. For the other direction, given $f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$, we write

$$A(n) \;:\equiv\; \Pi(\alpha, \beta\colon \mathbf{2}^{\mathbb{N}}).\ \alpha =_n \beta \to f\alpha = f\beta$$

for $n\colon \mathbb{N}$. Equality of natural numbers is a proposition (see [72, §3.1]), and hence so is $A(n)$, because, by function extensionality, a product of a family of propositions is a proposition (see [72, §3.6]). To conclude that $A(n)$ implies $A(m)$ decidable for all $m < n$, it is enough to show that for all n, if $A(n+1)$ holds then $A(n)$ is decidable. For every n, the type

$$B(n) = \Pi(s\colon \mathbf{2}^n).\ f(s0^\omega) = f(s1^\omega),$$

is decidable, because $\mathbb{N}$ has decidable equality and finite products of decidable types are also decidable. Now let $n : \mathbb{N}$ and assume $A(n+1)$. To show that $A(n)$ is decidable, it is enough to show that $A(n)$ is logically equivalent to $B(n)$, because then $B(n) \to A(n)$ and $\neg B(n) \to \neg A(n)$ and hence we can decide $A(n)$ by reduction to deciding $B(n)$.

The implication $A(n) \to B(n)$ holds without considering the assumption $A(n+1)$. To see this, assume $A(n)$ and let $s : \mathbf{2}^n$. Taking $\alpha = s0^\omega$ and $\beta = s1^\omega$, we conclude from $A(n)$ that $f(s0^\omega) = f(s1^\omega)$, which is the conclusion of $B(n)$.

Now assume $A(n+1)$ and $B(n)$. To establish $A(n)$, let $\alpha, \beta : \mathbf{2}^{\mathbb{N}}$ with $\alpha =_n \beta$. We need to conclude that $f(\alpha) = f(\beta)$. By the decidability of equality of $\mathbf{2}$, either $\alpha(n) = \beta(n)$ or not. If $\alpha(n) = \beta(n)$, then $\alpha =_{n+1} \beta$, and hence $f(\alpha) = f(\beta)$ by the assumption $A(n+1)$. If $\alpha_n \neq \beta_n$, we can assume w.l.o.g. that $\alpha_n = 0$ and $\beta_n = 1$. Now take $s = \alpha_0\alpha_1 \dots \alpha_{n-1} (= \beta_0\beta_1 \dots \beta_{n-1})$. Then $\alpha =_{n+1} s0^\omega$ and $s1^\omega =_{n+1} \beta$, which together with $A(n+1)$ imply $f(\alpha) = f(s0^\omega)$ and $f(s1^\omega) = f(\beta)$. But $f(s0^\omega) = f(s1^\omega)$ by $B(n)$, and hence $f(\alpha) = f(\beta)$ by transitivity.

Then Lemma 2.3.4 gives the desired result. □

Hence the Curry–Howard interpretation of (UC) is validated in the topological topos, even though the Curry–Howard interpretation of (Cont) fails in any topos.

2.4 Discussion

Thanks to the logical equivalence proved in Theorem 2.3.1, in a type theory with propositional truncation we can work with either formulation of the uniform-continuity principle. When we formalize in type theory the developments of Chapters 3–6, in Chapter 7, we will take (CH-UC), as this allows us to avoid the inclusion of propositional truncation, which is not available in MLTT. In Chapters 3–6 we will work in a deliberately informal approach to constructive mathematics, similar to Bishop's [10] in style, and we will speak of sets, that will be formalized as types in Chapter 7. It is worth emphasizing, however, that although we do not work with propositional truncation in Chapter 7, we found it essential to work with propositions in the above sense (namely types with at most one element, or sub-singletons) in order to make some of the informal arguments work in type theory. For example, in the official definition of (CH-UC) in the type-theoretic development (and in the Agda formalization), we require the existence of a minimal modulus of uniform continuity to make (CH-UC) into a proposition. Further discussion is in Chapter 7.1.

If we were considering the principle (Cont) as well, we would be forced to consider the extension of type theory with propositional truncation, as discussed above, in order

to consistently formulate it as

$$\Pi(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}).\ \Pi(\alpha{:}\mathbb{N}^{\mathbb{N}}).\ \|\Sigma(m{:}\mathbb{N}).\ \Pi(\beta{:}\mathbb{N}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta\|.$$

In view of Theorem 2.1.1, it is not possible to remove the truncation operation via the use of minimal moduli of continuity.

CHAPTER 3

A VARIATION OF THE TOPOLOGICAL TOPOS

As mentioned in previous chapters, the uniform-continuity principle (UC) is validated in Johnstone's topological topos [48], which is briefly recalled in Section 3.1 below. However, the treatment of the topological topos is non-constructive in several respects that are relevant to our work, in particular modelling (UC). For example, [48, Section 3] uses arguments by contradiction or case analysis via excluded middle in order to obtain an explicit description of the canonical coverage.

We work with a variation of the topological topos that allows us to constructively model a dependent type theory with (the Curry–Howard interpretation of) the uniform-continuity principle discussed in Chapter 2. In our variation, instead of working with the canonical coverage, we work with a simpler, explicitly given coverage that is suitable for modelling (UC) and still retains the "topological" character of the resulting topos.

Our model can be developed in a minimalistic constructive meta-theory, and has been implemented in intensional Martin-Löf type theory [61] using Agda notation [11, 12, 63] (Chapters 7 and 8). This enables us to extract computational content from type-theoretic proofs that use (UC), avoiding non-constructive arguments in the correctness of the computational extraction process. In this chapter, we work within informal constructive mathematics, along the lines of Bishop mathematics [10]. In Chapter 7, we discuss how the model can be developed within MLTT itself, with a development which we carried out in Agda [76]. A major difference between the development of this Chapter with that of Chapter 7 is that type theory is "proof relevant", and hence some of the definitions and arguments of this chapter have to be adapted to cope with the additional information that a type-theoretic development has to take into account. This is explained further in Chapter 7.

From Section 3.2 to Section 3.5, we (implicitly) use the axiom of choice to get moduli of uniform continuity of maps $\mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ and moduli of local constancy of maps $\mathbf{2}^{\mathbb{N}} \to X$. This is not a problem in MLTT, provided the existential quantifier $\exists$ is interpreted as the type former Σ. In a setting without choice, we would need to define uniform continuity and local constancy by explicitly requiring a modulus.

In this thesis we explore only the aspects of the topos that are necessary for the purpose of modelling (UC). In particular, it would be interesting to investigate how our sub-canonical coverage differs from the canonical one. As far as simple types are concerned, there is no difference, at least assuming classical logic in the meta-language, as we show in Chapter 4.2 that the interpretations of the simple types in our topos are (a category

equivalent to that of) Kleene–Kreisel spaces, like in the topological topos. More generally, we show in Chapter 4.2 that the Kuratowski limit spaces are fully embedded in our topos, in the same way as in the topological topos.

This chapter is organized as follows. We begin with recalling the development of the topological topos and some of its properties that are relevant to our work in Section 3.1. Then, in Section 3.2, we define our variation which consists of sheaves on a certain uniform-continuity site that is suitable for predicative, constructive analysis. In Section 3.3, we look at the full subcategory of concrete sheaves, and illustrate how they can be conveniently regarded as spaces, more precisely, as a variation of Spanier's quasi-topological spaces [68]. In Section 3.4, we investigate the representable sheaf, which is concrete and has the universal property of $\mathbf{1}+\mathbf{1}$ to the power the natural numbers object in the subcategory of concrete sheaves and in the sheaf topos. In Section 3.5, without using classical logic or Brouwerian principles [7], we construct a fan functional $(\mathbf{2}^{\mathbb{N}} \to \mathbb{N}) \to \mathbb{N}$ in the category of concrete sheaves, which continuously calculates least moduli of uniform continuity of maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$.

3.1 Johnstone's topological topos

We recall, in this section, the main aspects of Johnstone's topological topos [48] that are relevant to our investigation.

Let $\mathbb{N}_\infty$ be the set $\mathbb{N} \cup \{\infty\}$. We give it the usual topology that makes it into the one-point compactification of natural numbers: a subset $U \subseteq \mathbb{N}_\infty$ is open if and only if whenever it contains ∞ there exists $k \in \mathbb{N}$ such that all $n \geq k$ are in U.

Remark. It is folklore in constructive mathematics that a better behaved construction of $\mathbb{N}_\infty$ is as the set of decreasing binary sequences (or alternatively as the isomorphic set of binary sequences with at most one 1). This set is isomorphic to $\mathbb{N} \cup \{\infty\}$ if and only if LPO holds [32]. But such constructivity issues are unimportant in this section, as the treatment of the topological topos is already non-constructive.

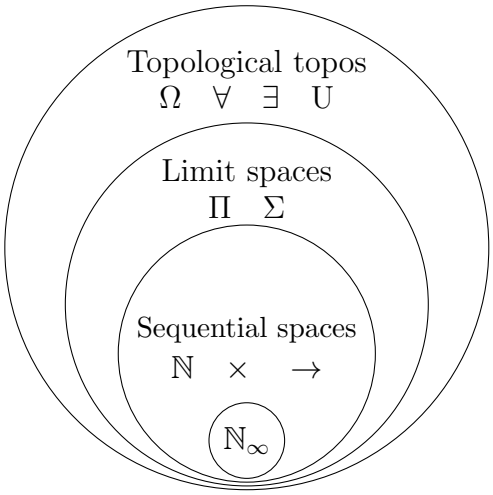
To build the topos, one takes the full subcategory $\mathbf{\Sigma}$ of the category **Top** of topological spaces, which has two objects: the one-point space $\mathbf{1}$ and the one-point compactification $\mathbb{N}_\infty$. Then the *topological topos* is the category $\mathbf{Shv}(\mathbf{\Sigma}, \mathcal{G})$ of sheaves on the site $(\mathbf{\Sigma}, \mathcal{G})$, where $\mathcal{G}$ is the canonical Grothendieck topology (see [59] for the relevant definitions).

As mentioned earlier, [48] also gives an explicit description of the canonical topology $\mathcal{G}$ as follows: $\mathcal{G}(\mathbf{1})$ is the set consisting of the maximal sieve on $\mathbf{1}$; and $\mathcal{G}(\mathbb{N}_\infty)$ is the set of sieves R on $\mathbb{N}_\infty$ such that (i) every map $\mathbf{1} \to \mathbb{N}_\infty$ is in R, and (ii) for every infinite $T \subseteq \mathbb{N}_\infty$, there exists an infinite $U \subseteq T$ such that $f_U \in R$, where $f_U \colon \mathbb{N}_\infty \to \mathbb{N}_\infty$ is the unique order-preserving monomorphism whose image is $U \cup \infty$. To prove that $\mathcal{G}$ is a Grothendieck topology and that $\mathcal{G}$ coincides with the canonical topology, arguments by contradiction or case analysis via excluded middle are employed in [48].

In fact, taking $\mathbf{\Sigma}$ to be the monoid of continuous endomaps of $\mathbb{N}_\infty$ would give an equivalent topos, but [48] finds it more convenient to consider $\mathbf{1}$ as a separate object in $\mathbf{\Sigma}$. For instance, with $\mathbf{1}$ in $\mathbf{\Sigma}$, the objects of the topological topos have a more intuitive view: if X is an object, then the elements of $X(\mathbf{1})$ are points, and the elements of $X(\mathbb{N}_\infty)$ are "proofs" that a given sequence of points converges.

The topological topos fully embeds some convenient categories of spaces, as indicated

in the following picture:



For any topological space X, the restricted hom-functor $\hom_{\mathbf{Top}}(-, X)\colon \Sigma^{\mathrm{op}} \to \mathbf{Set}$ is a sheaf over the site $(\Sigma, \mathcal{G})$. The assignment $X \mapsto \hom_{\mathbf{Top}}(-, X)$ gives a faithful functor $\mathbf{Top} \to \mathbf{Shv}(\Sigma, \mathcal{G})$. When restricted to the subcategory **Seq** of sequential spaces, this functor is full; thus, **Seq** can also be regarded as a full subcategory of the topological topos. Moreover, it is cartesian closed and has a natural numbers object, and hence serves as a model of Gödel's system T.

But sequential spaces are not enough to model dependent types, as the category **Seq** is not locally cartesian closed. Going beyond topological spaces, we have limit spaces, recalled in Chapter 4.1, which happen to be the subobjects of sequential spaces [48]. One can easily view the embedding of limit spaces in the topological topos as follows: the points are the same, and the convergent sequences are the proofs of convergence. In fact, limit spaces correspond to concrete sheaves [3] in the topological topos (see Section 3.3.1). Moreover, limit spaces form a locally cartesian closed category, which allows to interpret the type formers Π of dependent products and Σ of dependent sums [66].

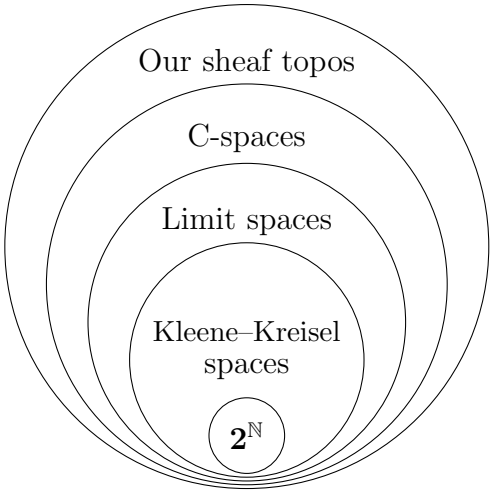
Using Streicher's construction [70] for arbitrary Grothendieck toposes, if we assume a Grothendieck universe in set theory, then we can build an object that serves as a Martin-Löf universe. However, this object is not a limit space (but it is an indiscrete object, in the sense that all maps into the Sierpinski space are constant, and hence it has only two "open subsets" [34]). Similarly, the subobject classifier Ω is also an object that fails to be a space but is indiscrete in the same sense, and hence the "logic" of the topos takes place in the "non-spatial", outer layer. But, because we work with the Curry-Howard interpretation of (UC), as discussed in Chapter 2, we can conveniently work in the layer of limit spaces. The same will apply to our variation of the topos, where an additional layer of C-spaces, containing limit spaces, arises.

Lastly, the Yoneda embedding sends the space $\mathbb{N}_\infty$ to a sheaf in the topological topos that corresponds to a limit space and a sequential space. Thus the one-object category containing $\mathbb{N}_\infty$ lives as a full subcategory of each of the above categories.

3.2 Our variation of the topological topos

Our variation of the topological topos consists of sheaves on a certain uniform-continuity site defined in Section 3.2.1. We investigate the cartesian closed structure of the topos

in Section 3.2.3. In particular, we look at concrete sheaves, which form an exponential ideal of the topos (by virtue of the equivalence of the categories of concrete sheaves and extensional sheaves) in Section 3.2.4.



The concrete sheaves can be conveniently regarded as spaces, which we call C-spaces, and their natural transformations can be regarded as continuous maps, as explored in the next section. Our C-spaces are analogous to limit spaces. For instance, they also form a (locally) cartesian closed category with a natural numbers object, and thus suffice to model simple types (Chapter 5) and even dependent types without universes (Chapter 6). Furthermore, in Chapter 4.2 we show that limit spaces are fully embedded within the category of C-spaces. Since the Kleene–Kreisel spaces live in the category of limit spaces, they can also be directly calculated within the one of C-spaces, as proved in Chapter 4.2. The Yoneda lemma maps the only object of the site to the *internal Cantor space* $\mathbf{2}^{\mathbb{N}}$, that is, the exponential of $\mathbf{2}$ to the power $\mathbb{N}$ in the topos, where $\mathbf{2}$ is $1 + 1$ and $\mathbb{N}$ is the natural numbers object. This allows us to construct a *fan* functional, which continuously calculates moduli of uniform continuity of maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$, using the Yoneda Lemma (Section 3.5).

3.2.1 The uniform-continuity site

As discussed above, the site of definition for the topological topos can be taken to be the monoid of continuous endofunctions of $\mathbb{N}_\infty$ equipped with the canonical Grothendieck topology. We replace this monoid by that of uniformly continuous endomaps of $\mathbf{2}^{\mathbb{N}}$, and the canonical converge by a subcanonical one, consisting of countably many finite covering families, which is suitable for predicative, constructive reasoning.

Let C be the monoid of *uniformly continuous* endomaps of the Cantor space $\mathbf{2}^{\mathbb{N}}$, that is, functions $t\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ such that

$$\forall m \in \mathbb{N}.\ \exists n \in \mathbb{N}.\ \forall \alpha, \beta \in \mathbf{2}^{\mathbb{N}}.\ \alpha =_n \beta \Rightarrow t\alpha =_m t\beta.$$

We write 1 for the identity map of $\mathbf{2}^{\mathbb{N}}$ as it is the identity element of the monoid C. Notice that any continuous function $\mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ is uniformly continuous, assuming classical logic or the Fan Theorem. Because we do not assume such principles, we need to explicitly require uniform continuity in the definition of the monoid C.

Our *coverage* $\mathcal{J}$ on the monoid C consists of the covering families

$$\{\mathrm{cons}_s\}_{s\in\mathbf{2}^n}$$

for all $n \in \mathbb{N}$, where $\mathbf{2}^n$ is the set of binary sequences of length n, and $\mathrm{cons}_s \colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ is the concatenation map:

$$\mathrm{cons}_s(\alpha) = s\alpha.$$

It is easy to verify that, for any $n \in \mathbb{N}$ and for any $s \in \mathbf{2}^n$, the map cons_s is uniformly continuous and thus an element of the monoid C. This subcanonical coverage $\mathcal{J}$ has the following convenient properties:

(1) $\mathcal{J}$ is countable.

(2) Each covering family is finite.

(3) Each covering family is jointly surjective.

(4) The maps in each covering family have disjoint images; thus, the compatibility condition the definition of sheaf holds automatically and hence can be ignored.

Because of (1), the *coverage axiom* specialized to our situation amounts to saying that, for all $t \in \mathrm{C}$,

$$\forall m \in \mathbb{N}.\ \exists n \in \mathbb{N}.\ \forall s \in \mathbf{2}^n.\ \exists t' \in \mathrm{C}.\ \exists s' \in \mathbf{2}^m.\ t \circ \mathrm{cons}_s = \mathrm{cons}_{s'} \circ t'. \qquad (\dagger)$$

Moreover, we have the following:

Lemma 3.2.1. *A map $t \colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ satisfies the coverage axiom (†) if and only if it is uniformly continuous.*

Proof. It is enough to prove

$$(\forall \alpha, \beta \in \mathbf{2}^{\mathbb{N}}.\ \alpha =_n \beta \Rightarrow t\alpha =_m t\beta) \iff (\forall s \in \mathbf{2}^n.\exists t' \in \mathrm{C}.\exists s' \in \mathbf{2}^m.t \circ \mathrm{cons}_s = \mathrm{cons}_{s'} \circ t')$$

for any $m, n \in \mathbb{N}$.

($\Rightarrow$) Given $s \in \mathbf{2}^n$, we define $t' \colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ by taking $t'\alpha$ to be the suffix of $t(s\alpha)$ after the first m bits for all $\alpha \in \mathbf{2}^{\mathbb{N}}$, and define $s' \in \mathbf{2}^m$ to be the prefix of $t(s0^\omega)$ of length m. Then the assumption $\forall \alpha, \beta \in \mathbf{2}^{\mathbb{N}}.\ \alpha =_n \beta \Rightarrow t\alpha =_m t\beta$ gives the required equation.

($\Leftarrow$) Given $\alpha, \beta \in \mathbf{2}^{\mathbb{N}}$, if $\alpha =_n \beta$ then they have the same prefix $s \in \mathbf{2}^n$ of length n. By the assumption we get $s' \in \mathbf{2}^m$ which is the prefix of $t\alpha$ and $t\beta$ of length m. □

Thus, not only does the coverage axiom hold, but also it amounts to the fact that the elements of the monoid C are the uniformly continuous functions. In virtue of this view, we call $\mathcal{J}$ the *uniform-continuity coverage* and $(\mathrm{C}, \mathcal{J})$ the *uniform-continuity site*.

As discussed earlier, using the axiom of choice, for each $t \in \mathrm{C}$ we can define a map $\mathrm{mod}_t \colon \mathbb{N} \to \mathbb{N}$, called the *modulus of uniform continuity*, such that

$$\forall m \in \mathbb{N}.\ \forall \alpha, \beta \in \mathbf{2}^{\mathbb{N}}.\ \alpha =_{\mathrm{mod}_t(m)} \beta \implies t\alpha =_m t\beta.$$

And, thanks to Lemma 3.2.1, it also satisfies

$$\forall m \in \mathbb{N}.\ \forall s \in \mathbf{2}^{\mathrm{mod}_t(m)}.\ \exists t' \in \mathrm{C}.\ \exists s' \in \mathbf{2}^m.\ t \circ \mathrm{cons}_s = \mathrm{cons}_{s'} \circ t'.$$

With a similar algorithm as that in the proof of Theorem 3.5.2, if t is uniformly continuous, then for any m we can always find the least n satisfying (†). This allows us to define the *least modulus of uniform continuity*, $\mathrm{lmod}_t \colon \mathbb{N} \to \mathbb{N}$, of the map t.

As mentioned before, our variation of the topological topos is the category $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$ of sheaves on the uniform-continuity site. Recall that a presheaf on a one-object category, i.e. a monoid, can be formulated in terms of monoid actions [59, §I.1]: A *presheaf* on C amounts to a set P with an *action*

$$((p, t) \mapsto p \cdot t) \colon P \times \mathrm{C} \to P$$

such that for all $p \in P$ and $t, r \in \mathrm{C}$

$$p \cdot 1 = p, \qquad p \cdot (t \circ r) = (p \cdot t) \cdot r.$$

Then a *natural transformation* of presheaves $(P, \cdot)$ and $(Q, \cdot)$ amounts to a map $\phi \colon P \to Q$ that preserves the action, i.e.

$$\phi(p \cdot t) = (\phi\ p) \cdot t$$

for all $p \in P$ and $t \in \mathrm{C}$.

Because the maps in each covering family have disjoint images, the amalgamation condition for a sheaf does not need to mention the compatibility condition in our case:

Lemma 3.2.2. *A presheaf $(P, \cdot)$ is a* sheaf *over $(\mathrm{C}, \mathcal{J})$ if and only if for any $n \in \mathbb{N}$ and $\{p_s \in P\}_{s \in \mathbf{2}^n}$, there is a unique* amalgamation *$p \in P$ such that, for all $s \in \mathbf{2}^n$,*

$$p \cdot \mathrm{cons}_s = p_s.$$

Notice also that, by induction, it is enough to consider the case $n = 1$:

Lemma 3.2.3. *A presheaf $(P, \cdot)$ is a* sheaf *over $(\mathrm{C}, \mathcal{J})$ if and only if for any $p_0, p_1 \in P$, there is a unique* amalgamation *$p \in P$ such that*

$$p \cdot \mathrm{cons}_0 = p_0 \qquad \textit{and} \qquad p \cdot \mathrm{cons}_1 = p_1.$$

We also call the above lemmas/definitions the *sheaf condition*. The first one is more convenient to work with when a sheaf is given, while the second one makes verifying the sheaf condition simpler.

To improve the readability of the thesis, we abbreviate P for the (pre)sheaf $(|P|, \cdot)$ where $|P|$ is the underlying set and $\cdot$ is its action, and we often write P to mean $|P|$ by an abuse of notation.

3.2.2 Subcanonicity of the uniform-continuity coverage

One example of a presheaf is the monoid C itself with function composition as its action. We have that the *Yoneda embedding* maps the only object $\star$ of the monoid C to this presheaf:

$$y(\star) = (\mathrm{C}, \circ).$$

The monoid C can be regarded as a one-object subcategory of that of topological spaces, with object $\mathbf{2}^{\mathbb{N}}$ and the morphisms all uniformly continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$, so that we can write the above equation as

$$y(\mathbf{2}^{\mathbb{N}}) = (\mathrm{C}, \circ).$$

Moreover, this presheaf is a sheaf. We only need to verify the sheaf condition: given any $t_0, t_1 \in \mathrm{C}$, the unique amalgamation $t\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ is simply

$$t(i\alpha) = t_i(\alpha),$$

whose uniform continuity is immediate. Hence every representable presheaf is a sheaf, which we record as follows:

Lemma 3.2.4. *The uniform-continuity coverage is subcanonical.*

This sheaf plays an important role in our model $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$, where it is used to construct exponentials (Section 3.2.3) and the fan functional (Section 3.5).

3.2.3 The cartesian closed structure of $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$

It is well known that the category of sheaves enjoys many convenient categorical properties [59, §III]. However, in this thesis, we are mainly working with its subcategory of C-spaces (Section 3.3), e.g. providing a constructive substitute for Kleene–Kreisel continuous functionals (Chapter 4) and modelling simple types with (UC) (Chapter 5). C-spaces correspond to concrete sheaves which are explored in Section 3.2.4. Here we briefly recall the cartesian closed structure of the category $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$, in order to understand its relation to concrete sheaves/C-spaces (see Corollaries 3.2.11 and 3.3.3).

Any singleton set $\mathbf{1} = \{\star\}$ works as a *terminal object* in $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$. Specifically, its action is defined by

$$\star \cdot t = \star$$

for all $t \in \mathrm{C}$. It is obvious that $\mathbf{1}$ satisfies the sheaf axiom, and that, for any sheaf P, the unique map $\mathrm{unit}\colon P \to \mathbf{1}$ is a natural transformation.

Given sheaves P and Q, the *product* $P \times Q$ is defined to be the set of all pairs (p, q) for $p \in P$ and $q \in Q$, with an action defined componentwise, i.e.

$$(p, q) \cdot t = (p \cdot t, q \cdot t)$$

for any $p \in P$, $q \in Q$ and $t \in \mathrm{C}$. It is easy to check that $P \times Q$ is a sheaf and the categorical product in $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$.

Given sheaves P and Q, we define the *exponential* Q^P to be the set of all natural transformations from $\mathrm{C} \times P$ to Q, with an action defined by

$$(\phi \cdot t)(r, p) = \phi(t \circ r, p)$$

for $\phi \in Q^P$, $t, r \in \mathrm{C}$ and $p \in P$. To prove that Q^P is a sheaf, we need the following:

Lemma 3.2.5. *Let Q be a sheaf and $q, q' \in Q$. The equation $q = q'$ holds if and only if there exists $n \in \mathbb{N}$ such that $q \cdot \mathrm{cons}_s = q' \cdot \mathrm{cons}_s$ holds for all $s \in \mathbf{2}^n$.*

Proof. ($\Rightarrow$) This direction follows from the definition. ($\Leftarrow$) Assume $q \cdot \mathrm{cons}_s = q' \cdot \mathrm{cons}_s$ for some $n \in \mathbb{N}$ and for all $s \in \mathbf{2}^n$. Then q is an amalgamation of the family $\{q' \cdot \mathrm{cons}_s\}_{s \in \mathbf{2}^n}$ and so is q'. As Q is a sheaf, there is only one unique amalgamation and thus $q = q'$. □

Proposition 3.2.6. *If P is a presheaf and Q is a sheaf, then Q^P is sheaf.*

Proof. Given $m \in \mathbb{N}$ and $\{\phi_s \in Q^P\}_{s \in \mathbf{2}^m}$, we construct an amalgamation ϕ as follows:

Given $r \in \mathrm{C}$, let $n = \mathrm{lmod}_r(m)$, i.e. the least modulus of uniform continuity of r. Then, for each $s \in \mathbf{2}^n$, we get $r_s \in \mathrm{C}$ and $s' \in \mathbf{2}^m$ such that $r \circ \mathrm{cons}_s = \mathrm{cons}_{s'} \circ r_s$. Now given $p \in P$, the family $\{\phi_{s'}(r_s, p \cdot \mathrm{cons}_s)\}_{s \in \mathbf{2}^n}$ has a unique amalgamation q, since $\phi_{s'}$ is given and Q is a sheaf. We define $\phi(r, p) = q$ and then have

$$\forall r \in \mathrm{C}.\ \forall p \in P.\ \forall s \in \mathbf{2}^{\mathrm{lmod}_r(m)}.\ \exists s' \in \mathbf{2}^m.\ \exists r_s \in \mathrm{C}.\ \phi(r, p) \cdot \mathrm{cons}_s = \phi_{s'}(r_s, p \cdot \mathrm{cons}_s). \quad (\ddagger)$$

We firstly show that ϕ is an amalgamation: for any $s \in \mathbf{2}^m$, $r \in \mathrm{C}$ and $p \in P$, we have

$$\begin{array}{lll} & (\phi \cdot \mathrm{cons}_s)(r, p) & \\ = & \phi(\mathrm{cons}_s \circ r, p) & \text{(by the action on } Q^P) \\ = & \phi(\mathrm{cons}_s \circ r, p) \cdot \mathrm{cons}_\varepsilon & \\ = & \phi_s(r, p \cdot \mathrm{cons}_\varepsilon) & \text{(by } (\ddagger) \text{ as } \mathrm{lmod}_{\mathrm{cons}_s \circ r}(m) = 0) \\ = & \phi_s(r, p). & \end{array}$$

This ϕ is unique: let ϕ' be another amalgamation. For any $r \in \mathrm{C}$, $p \in P$ and $s \in \mathbf{2}^{\mathrm{lmod}_r(m)}$, we have

$$\begin{array}{lll} & \phi'(r, p) \cdot \mathrm{cons}_s & \\ = & \phi'(r \circ \mathrm{cons}_s, p \cdot \mathrm{cons}_s) & \text{(by the naturality of } \phi') \\ = & \phi'(\mathrm{cons}_{s'} \circ r_s, p \cdot \mathrm{cons}_s) & \text{(by } (\dagger),\ \exists s' \in \mathbf{2}^m.\ \exists r_s \in \mathrm{C}.\ r \circ \mathrm{cons}_s = \mathrm{cons}_{s'} \circ r_s) \\ = & \phi_{s'}(r_s, p \cdot \mathrm{cons}_s) & (\phi' \text{ is an amalgamation}) \\ = & \phi(r, p) \cdot \mathrm{cons}_s & \text{(by } (\ddagger)) \end{array}$$

and thus $\phi'(r, p) = \phi(r, p)$ by Lemma 3.2.5.

It remains to show that ϕ is a natural transformation. Given $r, t \in \mathrm{C}$ and $p \in P$, we let $n_r = \mathrm{lmod}_r(m)$ and $n_t = \mathrm{lmod}_t(n_r)$. Now given $s \in \mathbf{2}^{n_t}$, we get $s' \in \mathbf{2}^{n_r}$ and $t' \in \mathrm{C}$ such that $t \circ \mathrm{cons}_s = \mathrm{cons}_{s'} \circ t'$ by $(\dagger)$. Then we have

$$\begin{array}{lll} & \phi(r, p) \cdot t \cdot \mathrm{cons}_s & \\ = & \phi(r, p) \cdot \mathrm{cons}_{s'} \cdot t' & (t \circ \mathrm{cons}_s = \mathrm{cons}_{s'} \circ t') \\ = & \phi_{s''}(r_{s'}, p \cdot \mathrm{cons}_{s'}) \cdot t' & \text{(by } (\ddagger)) \\ = & \phi_{s''}(r_{s'} \circ t', p \cdot \mathrm{cons}_{s'} \cdot t') & \text{(by the naturality of } \phi_{s''}) \\ = & \phi_{s''}(r_{s'} \circ t', p \cdot t \cdot \mathrm{cons}_s) & (t \circ \mathrm{cons}_s = \mathrm{cons}_{s'} \circ t') \\ = & \phi(r \circ t, p \cdot t) \cdot \mathrm{cons}_s. & ((r \circ t) \circ \mathrm{cons}_s = \mathrm{cons}_{s''} \circ (r_{s'} \circ t')) \end{array}$$

and thus $\phi(r, p) \cdot t = \phi(r \circ t, p \cdot t)$ by Lemma 3.2.5. □

Lemma 3.2.7. *If P and Q are sheaves, then the sheaf Q^P has the universal property of an exponential.*

Proof. We define the evaluation morphism by

$$\mathrm{eval} \colon Q^P \times P \to Q \qquad \mathrm{eval}(\phi, p) = \phi(1, p)$$

where $1\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ is the identity map. It is a natural transformation: given $t \in \mathrm{C}$, we have $\mathrm{eval}(\phi, p) \cdot t = \phi(1, p) \cdot t = \phi(t, p \cdot t) = \phi \cdot t(1, p \cdot t) = \mathrm{eval}(\phi \cdot t, p \cdot t)$.

Given a sheaf R and a natural transformation $g\colon R \times P \to Q$, we define a map

$$\lambda g\colon R \to Q^P \qquad \lambda g(r)(t, p) = g(r \cdot t, p)$$

which is a natural transformation: given $u \in \mathrm{C}$, we have $(\lambda g(r) \cdot u)(t, p) = \lambda g(r)(u \circ t, p) = g(r \cdot u \cdot t, p) = \lambda g(r \cdot u)(t, p)$ for all $(t, p) \in \mathrm{C} \times P$.

For any $r \in R$ and $p \in P$, we have $\mathrm{eval}(\lambda g(r), p) = \lambda g(r)(1, p) = g(r, p)$ and thus the following diagram commutes.

$$\begin{array}{ccc} R \times P & & \\ {\scriptstyle \lambda g \times 1_P}\downarrow & \searrow^{g} & \\ Q^P \times P & \xrightarrow[\mathrm{eval}]{} & Q \end{array}$$

The uniqueness of λg is easy to verify. □

Now we conclude the following:

Theorem 3.2.8. *The category* $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$ *is cartesian closed.*

3.2.4 Concrete and Extensional sheaves

We say a (pre)sheaf is called *concrete* if its action is function composition [3]. Then all the elements in a concrete (pre)sheaf $(P, \circ)$ must be maps from the Cantor space to some set X. Concrete sheaves admit a more concrete description as sets with the additional structures given by their elements, as we will discuss in Section 3.3. We denote the full subcategory of concrete sheaves by $\mathbf{CShv}(\mathrm{C}, \mathcal{J})$.

We now consider a subcategory of $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$ which is equivalent to $\mathbf{CShv}(\mathrm{C}, \mathcal{J})$, and clearly forms an exponential ideal of $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$. We say a (pre)sheaf $(P, \cdot)$ is *extensional* iff equality on P is determined by points: for every $p, p' \in P$, we have that $p = p'$ if and only if $p \cdot c = p' \cdot c$ for every constant $c\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ in the monoid C. We write $\mathbf{EShv}(\mathrm{C}, \mathcal{J})$ for the category of extensional sheaves which has the following properties:

Proposition 3.2.9. (1) *The terminal sheaf* $\mathbf{1}$ *is extensional.*

(2) *If sheaves* P *and* Q *are extensional, then so is* $P \times Q$*.*

(3) *If* P *is a sheaf and* Q *is an extensional sheaf, then* Q^P *is extensional.*

Proof. (1) and (2) are trivial. Here we only prove (3): given $\phi, \phi' \in Q^P$ with $\phi \cdot c = \phi' \cdot c$ for all constants $c \in \mathrm{C}$, we want to show that ϕ and ϕ' are pointwise equal. Given $t \in \mathrm{C}$, constant $c \in \mathrm{C}$ and $p \in P$, we have

$$\begin{array}{lll} & \phi(t, p) \cdot c & \\ = & \phi(t \circ c, p \cdot c) & \text{(by the naturality of } \phi) \\ = & (\phi \cdot (t \circ c))(1, p \cdot c) & \text{(by the action on } Q^P) \\ = & (\phi' \cdot (t \circ c))(1, p \cdot c) & (t \circ c \text{ is constant}) \\ = & \phi'(t \circ c, p \cdot c) & \text{(by the action on } Q^P) \\ = & \phi'(t, p) \cdot c, & \text{(by the naturality of } \phi') \end{array}$$

and thus $\phi(t, p) = \phi'(t, p)$ because Q is extensional. □

Proposition 3.2.10. *The categories* $\mathbf{ESh}(\mathrm{C}, \mathcal{J})$ *and* $\mathbf{CSh}(\mathrm{C}, \mathcal{J})$ *are equivalent.*

Proof. ($\Leftarrow$) Every concrete sheaf $(P, \circ)$ is extensional: if $p, p' \in P$ and $p \circ c = p' \circ c$ for all constant $c \in \mathrm{C}$, then we have $p(\alpha) = (p \circ (\lambda\beta.\alpha))(0^\omega) = (p' \circ (\lambda\beta.\alpha))(0^\omega) = p'(\alpha)$ for all $\alpha \in \mathbf{2}^{\mathbb{N}}$. Hence we have a functor $\sigma\colon \mathbf{CShv}(\mathrm{C}, \mathcal{J}) \to \mathbf{EShv}(\mathrm{C}, \mathcal{J})$ sends concrete sheaves to themselves.

($\Rightarrow$) Given an extensional sheaf $(P, \cdot)$, we get a concrete sheaf $(\bar{P}, \circ)$ where the set $\bar{P}$ is defined by

$$\bar{P} \;:\equiv\; \{\lambda\alpha.(p \cdot \lambda\beta.\alpha) \mid p \in P\}$$

It is clear that $(\bar{P}, \circ)$ is a sheaf. If P and Q are extensional sheaves, then for each natural transformation $\phi\colon P \to Q$ we can define $\bar{\phi}\colon \bar{P} \to \bar{Q}$ by composition, i.e. $\bar{\phi}(\bar{p}) = \phi \circ \bar{p}$ for all $\bar{p} \in \bar{P}$. Therefore, we get a functor $\tau\colon \mathbf{EShv}(\mathrm{C}, \mathcal{J}) \to \mathbf{CShv}(\mathrm{C}, \mathcal{J})$.

To prove that the compositions $\sigma \circ \tau$ and $\tau \circ \sigma$ are naturally isomorphic to the identity functors, it is equivalent to prove that any extensional sheaf P is isomorphic to $\bar{P}$. Clearly the map sending $p \in P$ to $\bar{p} \equiv \lambda\alpha.(p \cdot \lambda\beta.\alpha) \in \bar{P}$ is surjective. It is also injective: given $p \neq p' \in P$, we assume $\bar{p} = \bar{p}'$. Then $\bar{p}(\alpha) = \bar{p}'(\alpha)$, i.e. $p \cdot \lambda\beta.\alpha = p' \cdot \lambda\beta.\alpha$ for all $\alpha \in \mathbf{2}^{\mathbb{N}}$. This is equivalent to $p \cdot c = p' \cdot c$ for all constant $c \in \mathrm{C}$ using function extensionality. Since P is extensional we have $p = p'$ which leads to a contradiction. Thus we have a bijection between P and $\bar{P}$. □

Since equivalences of categories preserve most categorical concepts and properties, e.g. limits and colimits [58], we have the following:

Corollary 3.2.11. *Concrete sheaves form an exponential ideal of* $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$.

3.3 C-spaces and continuous maps

As mentioned earlier, concrete sheaves admit a convenient description as spaces, and their natural transformations as continuous maps. More precisely, they are analogous to Spanier's quasi-topological spaces [68]. In this section, we firstly demonstrate how concrete sheaves can be regarded as a variation of quasi-topological spaces, and call the resulting objects C-spaces. Then we explore some properties of the category of C-spaces that are necessary for modelling simple and dependent types.

3.3.1 Concrete sheaves as a variation of quasi-topological spaces

One advantage of quasi-topological spaces over topological spaces, which is the main reason for Spanier's introduction of the notion of quasi-topological space, is that quasi-topological spaces form a cartesian closed category. This category serves as a model of system T and HA^ω that validates the uniform-continuity principle, assuming classical logic in the meta-language. Our concrete sheaves can be seen as analogues of quasi-topological spaces, admitting a constructive treatment.

Recall that a *quasi-topology* on a set X assigns to each compact Hausdorff space K a set $Q(K, X)$ of functions $K \to X$ such that:

(1) All constant maps $K \to X$ are in $P(K, X)$.

(2) If $t\colon K' \to K$ is continuous and $q \in Q(K, X)$, then $q \circ t \in Q(K', X)$.

(3) If $\{t_i \colon K_i \to K\}_{i \in I}$ is a finite, jointly surjective family and $q \colon K \to X$ is a map with $q \circ t_i \in Q(K_i, X)$ for every $i \in I$, then $q \in Q(K, X)$.

A *quasi-topological space* is a set endowed with a quasi-topology, and a *continuous map* of quasi-topological spaces X and Y is a function $f \colon X \to Y$ such that $f \circ q \in Q(K, Y)$ whenever $q \in Q(K, X)$.

For example, every topological space X can be associated with a quasi-topology such that $Q(K, X)$ is the set of continuous maps $K \to X$ for every compact Hausdorff space K. Then it follows that every continuous map $f \colon X \to Y$ of topological spaces X and Y is continuous in the associated quasi-topologies. Therefore, this construction gives the full embedding of topological spaces into quasi-topological spaces.

The definition of quasi-topological space can be modified by considering just one compact Hausdorff space, the Cantor space, rather than all compact Hausdorff spaces, and by restricting the jointly surjective finite families of continuous maps to the covering families $\{\mathrm{cons}_s\}_{s \in \mathbf{2}^n}$ considered in the previous section. Then we have the following:

Definition 3.3.1. A C-*space* is a set X equipped with a C-*topology* P, i.e. a collection of maps $\mathbf{2}^{\mathbb{N}} \to X$, called *probes*, satisfying the following conditions, called the *probe axioms*:

(1) All constant maps are in P.

(2) (*Presheaf condition*) If $p \in P$ and $t \in \mathrm{C}$, then $p \circ t \in P$.

(3) (*Sheaf condition*) For any $n \in \mathbb{N}$ and any family $\{p_s \in P\}_{s \in \mathbf{2}^n}$, the unique map $p \colon \mathbf{2}^{\mathbb{N}} \to X$ defined by $p(s\alpha) = p_s(\alpha)$ is in P.

A *continuous map* of C-spaces (X, P) and (Y, Q) is a map $f \colon X \to Y$ with $f \circ p \in Q$ whenever $p \in P$. We write C-**Space** for the category of C-spaces and continuous maps. The above three conditions are called the *probe axioms*.

Notice that the sheaf condition is logically equivalent to

(3′) For any $p_0, p_1 \in P$, the map $p \colon \mathbf{2}^{\mathbb{N}} \to X$ defined by $p(i\alpha) = p_i(\alpha)$ is in P.

and

(3″) If $p \colon \mathbf{2}^{\mathbb{N}} \to X$ is a map such that there exists $n \in \mathbb{N}$ with $p \circ \mathrm{cons}_s \in P$ for all $s \in \mathbf{2}^n$, then $p \in P$.

(3′) is a special case of (3) where $n = 1$, and is equivalent to (3) by induction on n. When verifying that a given set is a C-space, it is more convenient to use (3′). And (3″) is the uncurried result of (3), and is more convenient to use if one already knows that a given set is a C-space.

The idea is that we "topologize" the set X by choosing a designated set P of maps $\mathbf{2}^{\mathbb{N}} \to X$ that we want, and hence declare, to be continuous. For example, if X already has some form of topology, e.g. a metric, we can take P to be the set of continuous functions $\mathbf{2}^{\mathbb{N}} \to X$ with respect to this topology and the natural topology of the Cantor space. Of course we have to make sure the sheaf condition is satisfied.

As mentioned earlier, C-spaces provide a more concrete description of concrete sheaves in the following sense. Given a C-space (X, P), the C-topology P together with function composition is a concrete sheaf. Conversely, if $(P, \circ)$ is a concrete sheaf, then all maps in P should have the same codomain which is the underlying set of the resulted C-space.

Proposition 3.3.2. *The two categories* C-**Space** *and* $\mathbf{CShv}(\mathrm{C}, \mathcal{J})$ *are equivalent.*

By virtue of this equivalence, C-**Space** can also be viewed as a full subcategory of $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$. With the Corollary 3.2.11, we have the following:

Corollary 3.3.3. C*-spaces form an exponential ideal of* $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$.

Similarly to our abbreviation for (pre)sheaves in the previous section, we abbreviate X for the space $(|X|, \mathrm{Probe}(X))$, where $|X|$ stands for the underlying set and $\mathrm{Probe}(X)$ for the collection of probes, i.e. the C-topology on $|X|$, and often write X to mean $|X|$.

3.3.2 The (local) cartesian closed structure of C-Space

Here we explore the cartesian closed structure of the category C-**Space**, in order to model simple types (Chapter 5), as well as its local cartesian closed structure, in order to model dependent types (Chapter 6).

Theorem 3.3.4. *The category* C-**Space** *is cartesian closed.*

Proof. Any singleton set $\mathbf{1} = \{\star\}$ with the unique map $\mathbf{2}^{\mathbb{N}} \to \mathbf{1}$ as the only probe is clearly a C-space and a *terminal object* in C-**Space**.

Given C-spaces (X, P) and (Y, Q), their *product* is the cartesian product $X \times Y$ equipped with the C-topology R defined by the condition that $r \colon \mathbf{2}^{\mathbb{N}} \to X \times Y$ is in R iff $\mathrm{pr}_1 \circ r \in P$ and $\mathrm{pr}_2 \circ r \in Q$, where pr_1 and pr_2 are the projections. We skip the routine verifications of probe axioms and the required universal property.

Given C-spaces (X, P) and (Y, Q), their *exponential* is the set Y^X of continuous maps $X \to Y$ equipped with the C-topology R defined by the condition that $r \colon \mathbf{2}^{\mathbb{N}} \to Y^X$ is in R iff for any $t \in \mathrm{C}$ and $p \in P$ the map $\lambda\alpha.r(t\alpha)(p\alpha)$ is in Q. Here we verify only the sheaf condition (3″): Suppose $r \colon \mathbf{2}^{\mathbb{N}} \to Y^X$ is a map such that $r \circ \mathrm{cons}_s \in R$ for all $s \in \mathbf{2}^n$ for some $n \in \mathbb{N}$. Given $t \in \mathrm{C}$ and $p \in P$, let $m = \mathrm{mod}_t(n)$. Then for each $s \in \mathbf{2}^m$, we get $t' \in \mathrm{C}$ and $s' \in \mathbf{2}^n$ such that $t \circ \mathrm{cons}_s = \mathrm{cons}_{s'} \circ t'$. We have

$$(\lambda\alpha.r(t\alpha)(p\alpha)) \circ \mathrm{cons}_s = \lambda\alpha.r(t(\mathrm{cons}_s\alpha))(p(\mathrm{cons}_s\alpha)) = \lambda\alpha.r(\mathrm{cons}_{s'}(t'\alpha))(p(\mathrm{cons}_s\alpha)).$$

Since $r \circ \mathrm{cons}_{s'}$ is in R, the map $\lambda\alpha.r(\mathrm{cons}_{s'}(t'\alpha))(p(\mathrm{cons}_s\alpha))$ is in Q, and so is the map $(\lambda\alpha.r(t\alpha)(p\alpha)) \circ \mathrm{cons}_s$. Then the sheaf condition (3″) of Y gives the desired result. □

Colimits of sheaves are generally constructed as the sheafifications of the ones in the category of presheaves (see [59, §III.6]). Here we present a direct construction of finite coproducts of C-spaces.

Theorem 3.3.5. *The category* C-**Space** *has finite coproducts.*

Proof. The empty set equipped with the empty C-topology is clearly a C-space and an *initial object* in C-**Space**.

Binary coproducts can be constructed as follows: given C-spaces (X, P) and (Y, Q), their *coproduct* is the disjoint union $X + Y$ equipped with the C-topology R defined by the condition that $r \colon \mathbf{2}^{\mathbb{N}} \to X + Y$ is in R iff there exists $n \in \mathbb{N}$ such that for all $s \in \mathbf{2}^n$ either there exists $p \in P$ with $r(\mathrm{cons}_s\alpha) = \mathrm{inl}(p\alpha)$ for all $\alpha \in \mathbf{2}^{\mathbb{N}}$ or there exists $q \in Q$ with $r(\mathrm{cons}_s\alpha) = \mathrm{inr}(q\alpha)$ for all $\alpha \in \mathbf{2}^{\mathbb{N}}$, where inl and inr are the injections. Here we verify only the sheaf condition (3′): Given $r_0, r_1 \in R$, we get n_0 and n_1 from their witnesses of being a probe on $X + Y$. For the map $r \colon \mathbf{2}^{\mathbb{N}} \to X + Y$ defined by $r(i\alpha) = r_i(\alpha)$, one can clearly see that the maximum of n_0 and n_1 is the desired n that makes $r \in R$. □

The category C-**Space** has all pullbacks, which are constructed in the same way as in **Set**. An exponential in a slice category C-**Space**$/X$ is constructed in the same way as in the slice category **Set**$/X$, with a suitable construction of the C-topology on its domain. The proof is available in [3, Proposition 43], in the generality of concrete sheaves on concrete sites. Here we present the construction, but skip the verification as it is similar to the one for exponentials of C-spaces (which is present in our formalization).

Theorem 3.3.6. *The category* C-**Space** *is locally cartesian closed.*

Proof. We skill the easy constructions and verifications of a terminal object and products in a slice category, but give the construction of exponentials.

Given a continuous map $f\colon X \to Y$ and an element $y \in Y$, the *fiber*

$$f^{-1}(y) = \{x \in X \mid f(x) = y\}$$

is a C-space, whose C-topology is inherited from X.

Given objects $X \xrightarrow{f} Y$ and $Z \xrightarrow{g} Y$ in C-**Space**$/Y$, we construct the exponential g^f as follows: The underlying set of the domain of g^f is defined by

$$\mathrm{dom}(g^f) = \{(y, \phi) \mid y \in Y,\ \phi\colon f^{-1}(y) \xrightarrow{\mathrm{cts}} g^{-1}(y)\}$$

The C-topology on $\mathrm{dom}(g^f)$ is defined by the condition that a map $r\colon \mathbf{2}^{\mathbb{N}} \to \mathrm{dom}(g^f)$ is a probe iff

(i) the composite $\mathrm{pr}_1 \circ r\colon \mathbf{2}^{\mathbb{N}} \to Y$ is a probe on Y, and

(ii) for any $t \in \mathrm{C}$ and $p \in \mathrm{Probe}(X)$ such that $\forall \alpha \in \mathbf{2}^{\mathbb{N}}.\ \mathrm{pr}_1(r(t\alpha)) = f(p\alpha)$, the map $\lambda\alpha.\mathrm{pr}_2(r(t\alpha))(p\alpha)$ is a probe on Z.

Verifying the sheaf condition for $\mathrm{dom}(g^f)$ is similar to the one for exponentials in C-**Space**. The exponential $g^f : \mathrm{dom}(g^f) \to Y$ is then defined to be the first projection. Condition (i) amounts to the continuity of g^f. And the idea of (ii) is that the composite

$$\mathbf{2}^{\mathbb{N}} \times_Y X \xrightarrow{r \times 1_X} \mathrm{dom}(g^f) \times_Y X \xrightarrow{\mathrm{ev}} Z$$

is continuous, where evaluation map ev applies the second component of $(y, \phi) \in \mathrm{dom}(g^f)$ to $x \in f^{-1}(y) \subseteq X$. □

3.3.3 Discrete C-spaces and natural numbers object

We say that a C-space X is *discrete* if for every C-space Y, all functions $X \to Y$ are continuous. A map $p\colon \mathbf{2}^{\mathbb{N}} \to X$ into a set X is called *locally constant* iff

$$\exists m \in \mathbb{N}.\ \forall \alpha, \beta \in \mathbf{2}^{\mathbb{N}}.\ \alpha =_m \beta \implies p(\alpha) = p(\beta).$$

We call m the *modulus of local constancy* of p.

Lemma 3.3.7. *Let X be any set.*

(1) *The locally constant functions $\mathbf{2}^{\mathbb{N}} \to X$ form a* C*-topology on X.*

(2) *For any* C*-topology* P *on* X*, every locally constant function* $\mathbf{2}^{\mathbb{N}} \to X$ *is in* P*.*

Proof. (1) Let P be a collection of all locally constant maps into X. The first two probe axioms are obviously satisfied. We only verify the sheaf condition (3′): If p_0 and p_1 are locally constant with moduli m_0 and m_1, then the unique map $p\colon \mathbf{2}^{\mathbb{N}} \to X$ defined by $p(i\alpha) = p_i(\alpha)$ is locally constant with the modulus $\max(m_0, m_1) + 1$.

(2) Let $p\colon \mathbf{2}^{\mathbb{N}} \to X$ be locally constant and n be its modulus of local constancy. Then, for each $s \in \mathbf{2}^n$, the composite $p \circ \mathrm{cons}_s$ is constant and thus a probe on X. Using the sheaf condition, we know that p is a probe on X. □

In other words, the locally constant maps $\mathbf{2}^{\mathbb{N}} \to X$ form the finest C-topology on the set X, in the sense of the smallest collection of probes. Moreover:

Lemma 3.3.8. *A* C*-space is discrete if and only if the probes on it are precisely the locally constant functions.*

Proof. ($\Rightarrow$) Let (X, P) be a discrete C-space. According to the previous lemma, all locally constant maps $\mathbf{2}^{\mathbb{N}} \to X$ form a C-topology, say Q, on X. Because (X, P) is discrete, the map $(X, P) \to (X, Q)$ which is identity on points is continuous. By the definition of continuity, all elements in P are also in Q, i.e. are locally constant.

($\Leftarrow$) Let P be the collection of all locally constant functions into X. Given a C-space (Y, Q) and a map $f\colon X \to Y$, we show that f is continuous: if $p\colon \mathbf{2}^{\mathbb{N}} \to X$ is locally constant whose modulus is n, then, for each $s \in \mathbf{2}^n$, the composite $f \circ p \circ \mathrm{cons}_s$ is constant and thus a probe on Y. By the sheaf condition, the map $f \circ p$ is a probe on Y. □

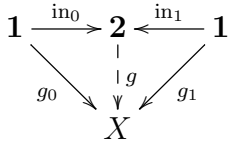
We thus refer to the collection of locally constant maps $\mathbf{2}^{\mathbb{N}} \to X$ as the *discrete* C-topology on X. In particular, when the set X is $\mathbf{2}$ or $\mathbb{N}$, the locally constant functions amount to the uniformly continuous functions. Hence we have a discrete two-point space $\mathbf{2}$ and a discrete space $\mathbb{N}$ of natural numbers, which play an important role in our model:

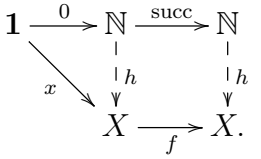
Theorem 3.3.9. *In the category* C-**Space***:*

(1) *The discrete two-point space* $\mathbf{2}$ *is the coproduct of two copies of the terminal space* $\mathbf{1}$*.*

(2) *The discrete space* $\mathbb{N}$ *of natural numbers is the natural numbers object.*

Proof. The universal properties of $\mathbf{2}$ and $\mathbb{N}$ can be constructed in the same way as in the category **Set**, because the unique maps g and h in the diagrams below are continuous by the discreteness of $\mathbf{2}$ and $\mathbb{N}$:





□

3.4 The representable sheaf is the Cantor space

Because our site is a one-object category, there is only one representable presheaf, up to isomorphism, which is a sheaf, as we have seen in Lemma 3.2.4 of Section 3.2.2. In this section we show that it is the exponential $\mathbf{2}^{\mathbb{N}}$ in the category C-**Space** and hence also in $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$. We exploit this in Section 3.5 below to construct a *fan functional* $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}} \to \mathbb{N}$ that continuously calculates minimal moduli of continuity in the category C-**Space**.

As discussed in Section 3.2.2, the Yoneda embedding $y\colon \mathrm{C} \to \mathbf{Shv}(\mathrm{C}, \mathcal{J})$ gives

$$y(\mathbf{2}^{\mathbb{N}}) = (\mathrm{C}, \circ),$$

where $\mathbf{2}^{\mathbb{N}}$ is the only object of the category C regarded as a subcategory of that of topological spaces, and where $(\mathrm{C}, \circ)$ is the sheaf with composition $\mathrm{C} \times \mathrm{C} \to \mathrm{C}$ as the action. This sheaf is concrete, and hence can be seen as a C-space, so that the Yoneda embedding restricts to a functor

$$y\colon \mathrm{C} \to \text{C-}\mathbf{Space}.$$

This concrete sheaf, seen as a C-space, as in Section 3.3.1, is the set $\mathbf{2}^{\mathbb{N}}$ equipped with all uniformly continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ as the probes. The following lemma shows that this C-space is the exponential $\mathbf{2}^{\mathbb{N}}$ of the discrete C-spaces $\mathbf{2}$ and $\mathbb{N}$ in the category C-space, as calculated in Theorem 3.3.4. Hence we can write

$$y(\mathbf{2}^{\mathbb{N}}) = \mathbf{2}^{\mathbb{N}},$$

where, as already discussed, $\mathbf{2}^{\mathbb{N}}$ in the left-hand side stands for the only object of the monoid C, and, in the right-hand side, for an exponential in the category of C-spaces. This notational overloading should cause no confusion.

Lemma 3.4.1. *The* C*-space* $y(\mathbf{2}^{\mathbb{N}})$ *has the universal property of* $\mathbf{2}$ *to the power* $\mathbb{N}$ *in the category* C-**Space**, *and hence also in the category* $\mathbf{Shv}(\mathrm{C}, \mathcal{J})$.

Proof. By the proof of Theorem 3.3.4, the underlying set of the exponential can be calculated as the set of all continuous maps $\mathbb{N} \to \mathbf{2}$, which, by the discreteness of $\mathbb{N}$, amounts to the set $\mathbf{2}^{\mathbb{N}}$ of all maps $\mathbb{N} \to \mathbf{2}$. Again by Theorem 3.3.4, the probes on the exponential are the maps $r\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ such that

$$\forall t \in \mathrm{C}.\ \forall p \in \mathrm{Probe}(\mathbb{N}).\ \lambda\alpha.r(t\alpha)(p\alpha) \in \mathrm{Probe}(\mathbf{2}). \qquad (\ddagger)$$

To conclude the proof, we have to show that the C-topologies of the C-space $y(\mathbf{2}^{\mathbb{N}})$ and of the exponential C-space $\mathbf{2}^{\mathbb{N}}$ coincide, which amounts to showing that a map r satisfies $(\ddagger)$ iff it is uniformly continuous:

$(\Rightarrow)$ Let $n \in \mathbb{N}$ be given. For each $i \leq n$, we take t to be the identity and $p = \lambda\alpha.i$, and have that $\lambda\alpha.(r\alpha)_i\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}$ is a probe and hence uniformly continuous. Let m_i be its modulus of uniform continuity, and let m be the maximum of $\{m_i\}_{i\leq n}$. Then, for any $\alpha, \beta \in \mathbf{2}^{\mathbb{N}}$ with $\alpha =_m \beta$, we have $(r\alpha)_i = (r\beta)_i$ for all $i \leq n$, *i.e.* $r\alpha =_n r\beta$.

$(\Leftarrow)$ Let $t \in \mathrm{C}$ and $p \in \mathrm{Probe}(\mathbb{N})$ be given. Because p is uniformly continuous, its image is finite. We take n to be the maximum and then have that $p\alpha \leq n$ for all $\alpha \in \mathbf{2}^{\mathbb{N}}$. Both r and t are uniformly continuous, and so is $r \circ t$. Thus we take $m = \mathrm{mod}_{r\circ t}(n)$, where

mod is the modulus-of-uniform-continuity function defined in Section 3.2.1. Let k be the maximum of n and m. Then, for any $\alpha, \beta \in \mathbf{2}^{\mathbb{N}}$ with $\alpha =_k \beta$, we have $r(t\alpha) =_n r(t\beta)$ by the uniform continuity of $r \circ t$, and $p\alpha = p\beta$ by the uniform continuity of p. As $p\alpha \leq n$, we have $r(t\alpha)(p\alpha) = r(t\beta)(p\beta)$. Hence $\lambda\alpha.r(t\alpha)(p\alpha)$ is uniformly continuous and thus a probe on $\mathbf{2}$. □

Using this, we conclude that the Yoneda Lemma amounts to saying that a function $\mathbf{2}^{\mathbb{N}} \to X$ into a C-space X is a probe iff it is continuous. More precisely:

Lemma 3.4.2 (Yoneda)**.** *A map of the set $\mathbf{2}^{\mathbb{N}}$ to the underlying set of a* C*-space X is a probe if and only if it is continuous when regarded as a map from the exponential $\mathbf{2}^{\mathbb{N}}$ to the space X in the category* C-**Space**.

Proof. ($\Rightarrow$) Let $p\colon \mathbf{2}^{\mathbb{N}} \to X$ be a probe on X. By the presheaf condition of X, we have that $p \circ t$ is a probe on X for each $t \in \mathrm{C}$, which means that the map p is continuous. ($\Leftarrow$) Let $p\colon \mathbf{2}^{\mathbb{N}} \to X$ be a continuous map. The identity map 1 is uniformly continuous and thus a probe on $\mathbf{2}^{\mathbb{N}}$. Thus $p = p \circ 1$ is a probe on X by the continuity of p. □

3.5 The fan functional in the category of C-spaces

Although the categories of topological spaces and of locales fail to be cartesian closed, they have the exponentials $\mathbf{2}^{\mathbb{N}}$ and $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$, where $\mathbf{2}$ is $1 + 1$ and $\mathbb{N}$ is the natural numbers object. Moreover, the exponential $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$ is discrete [47, 30]. The same phenomenon takes place in the category of C-spaces.

Lemma 3.5.1. *The exponential $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$ is a discrete* C*-space.*

Proof. Given a probe $p\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$, we want to show that it is locally constant. By the construction of exponentials in the previous section, we know that for all $t, r \in \mathrm{C}$,

$$\lambda\alpha.p(t\alpha)(r\alpha) \in \mathrm{Probe}(\mathbb{N}),$$

i.e. $\lambda\alpha.p(t\alpha)(r\alpha)$ it is uniformly continuous. In particular, we can take

$$t(\alpha)(i) = \alpha_{2i} \qquad \text{and} \qquad r(\alpha)(i) = \alpha_{2i+1},$$

both of which are uniformly continuous, and define $q\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ by $q(\alpha) = p(t\alpha)(r\alpha)$. From the uniform-continuity witness of q, we get its modulus n. Then we define a map $\mathrm{join}\colon \mathbf{2}^{\mathbb{N}} \times \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ by

$$\begin{array}{lcl} \mathrm{join}(\alpha, \beta)(2i) & = & \alpha_i \\ \mathrm{join}(\alpha, \beta)(2i+1) & = & \beta_i. \end{array}$$

Given $\alpha, \alpha', \beta \in \mathbf{2}^{\mathbb{N}}$ with $\alpha =_n \alpha'$, we have

$$\begin{array}{lll} & p(\alpha)(\beta) & \\ = & p(t(\mathrm{join}(\alpha, \beta)))(r(\mathrm{join}(\alpha, \beta))) & \text{(by the definitions of } t, r, \mathrm{join}) \\ = & q(\mathrm{join}(\alpha, \beta)) & \text{(by the definition of } q) \\ = & q(\mathrm{join}(\alpha', \beta)) & (\mathrm{join}(\alpha, \beta) =_{2n} \mathrm{join}(\alpha', \beta), 2n \geq n) \\ = & p(\alpha')(\beta). & \end{array}$$

Hence p is locally constant and therefore $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$ is discrete. □

As in the classical case of Kleene–Kreisel functionals, this lemma is at the heart of our construction of the fan function in the category C-**Space**. Notice, however, that we did not need to use a compactness argument, as is needed for Kleene–Kreisel functionals, which allowed us to avoid classical logic and Brouwer's Fan Theorem to prove the above lemma. We remark that Hyland [47] proves the above lemma in the category of locales also in intuitionistic logic without Brouwerian principles.

Theorem 3.5.2. *There is a* fan functional

$$\mathrm{fan}\colon \mathbb{N}^{\mathbf{2}^{\mathbb{N}}} \to \mathbb{N}$$

in C-**Space** *that continuously calculates least moduli of uniform continuity.*

Proof. Given $f \in \mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$, i.e. a continuous map $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$, we know f is uniformly continuous according to the Yoneda lemma. Then, form the witness of its uniform continuity, we get a modulus m_f.

From this modulus we calculate the least modulus of f as follows: We define a function $\mathrm{lmod}\colon (\mathbf{2}^{\mathbb{N}} \to \mathbb{N}) \to \mathbb{N} \to \mathbb{N}$ by induction on its second argument:

$$\begin{array}{lcl} \mathrm{lmod}\ f\ 0 & = & 0 \\ \mathrm{lmod}\ f\ (n+1) & = & \text{if } (\forall s \in \mathbf{2}^n.\ f(s0^\omega) = f(s1^\omega)) \quad \text{then} \quad (\mathrm{lmod}\ f\ n) \\ & & \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{else} \quad (n+1). \end{array}$$

With a proof by induction, we can show that $\mathrm{lmod}\ f\ n$ is the smallest modulus if n is a modulus of f. Hence, we define

$$\mathrm{fan}(f) = \mathrm{lmod}\ f\ m_f.$$

As the space $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$ is discrete by the previous lemma, this functional is continuous. □

We use this functional to model the uniform-continuity principle, which is expressed as a skolemized formula in system T (Chapter 5.1) and as a type via the Curry–Howard interpretation in MLTT (Chapter 6.2)

CHAPTER 4

The Kleene–Kreisel continuous functionals

From a constructive point of view, the traditional treatment of the Kleene–Kreisel continuous functionals [64, 65, 56, 57] is problematic, because the proofs available in the literature rely on either classical logic or constructively contentious principles such as Bar Induction or the Fan Theorem [7, 73]. One such example is the fact that all functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ in the category of Kleene–Kreisel functional are *uniformly* continuous. It is thus natural to ask whether it is possible to develop the theory of Kleene–Kreisel spaces constructively, which was one of the original motivations of this project. It turns out that this is indeed the case, using our category of C-spaces, constructively developed in Chapter 3.3, to host the Kleene–Kreisel spaces as a full subcategory: we start with the natural numbers object, and close under finite products and function spaces.

We start the chapter with a brief introduction to the Kleene–Kreisel continuous functionals and some approaches to them. In particular, we recall the development of limit spaces which host the Kleene–Kreisel spaces as a full subcategory, and some of their properties that are relevant to our investigation.

Then, in the next section, we show that limit spaces are fully embedded in the category of C-spaces. Moreover, the embedding becomes an equivalence when restricted to simple objects, that is the least collection containing the natural numbers object and closed under finite products and exponentials. Since the Kleene–Kreisel spaces are precisely the simple objects in the category of limit spaces, they can also be calculated within C-spaces.

We use non-constructive arguments in Section 4.2 (and in no other section) to establish the equivalence of Kleene–Kreisel spaces calculated within limit spaces and within C-spaces. The point is that our constructive development of C-spaces provides a classically equivalent substitute for the traditional manifestations of the Kleene–Kreisel spaces, which admits a constructive treatment of the uniform-continuity principle, as illustrated in Chapters 5 and 6.

In the last section, we show that the full type hierarchy is equivalent to the Kleene–Kreisel hierarchy within C-spaces under the assumption of the Brouwerian axiom that all set-theoretic functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous. Here the proof is constructive, and has been formalized in intensional MLTT in Agda notation [76]. It is interesting that other Brouwerian axioms such as more general forms of continuity or Bar Induction are not needed to prove the equivalence.

4.1 The Kleene–Kreisel continuous functionals

Continuous functionals were first discussed in Kleene's paper [50], using the notion of *countable functional*, and in Kreisel's paper [53], using the notion of *continuous functional*, and thus known as the *Kleene–Kreisel continuous functionals*. In both approaches, the aim is to develop a hierarchy of total functionals of finite types. The idea of this type hierarchy is that the action of a functional on an input is locally determined via finite approximation to that input.

In a cartesian closed category with a natural numbers object $\mathbb{N}$, define the *simple objects* to be the least collection containing $\mathbb{N}$ and closed under exponentials. The simple objects of any such category give an interpretation of the simply typed lambda calculus and higher-type primitive recursion (the term language of Gödel's system T). The Kleene–Kreisel *continuous functionals* form a category equivalent to the full subcategory on the simple objects of any of the following categories, among others:

1. compactly generated topological spaces [64, 33],
2. sequential topological spaces [33],
3. Simpson and Schröder's QCB spaces [4, 33],
4. Kuratowski limit spaces [45],
5. filter spaces [45],
6. Scott's equilogical spaces [5]
7. Johnstone's *topological topos* [48].

See Normann [65] and Longley [56, 57] for the relevance of Kleene–Kreisel spaces in the theory of higher-type computation. Counter-examples include Hyland's *effective topos* [46] and the hereditary effective operations (HEO) [56], which give a second simple-type hierarchy (see [56] for a discussion).

Here we briefly recall the definition of limit space and some properties that will be used in the next section. A *limit space* (also known as a *subsequential space* in [48] and an *L-space* in [45]) is a set X together with a family of functions $x\colon \mathbb{N}_\infty \to X$, written as $(x_i) \to x_\infty$ and called *convergent sequences* in X, satisfying the following conditions:

1. The constant sequence (x) converges to x.
2. If (x_i) converges to x_∞, then so does every subsequence of (x_i).
3. If (x_i) is a sequence such that every subsequence of (x_i) contains a subsequence converging to x_∞, then (x_i) converges to x_∞.

We call the collection of convergent sequences in X the *limit-structure* on X. A function $f\colon X \to Y$ of limit spaces is said to be *continuous* if it preserves convergent sequences, i.e. $(fx_i) \to fx_\infty$ whenever $(x_i) \to x_\infty$. We write **Lim** to denote the category of limit spaces and continuous maps.

As mentioned in Chapter 3.1, the category **Lim** is cartesian closed and has a natural numbers object. Here we give the constructions of products and exponentials of limit

spaces, omitting their verification: Let X and Y be limit spaces. The underlying set of the *product* $X \times Y$ consists of pairs of elements of X and Y; and, a sequence (x_i, y_i) converges to (x, y) in $X \times Y$ iff $(x_i) \to x$ in X and $(y_i) \to y$ in Y. The underlying set of the *exponential* Y^X consists of continuous maps $X \to Y$; and, a sequence (f_i) converges to f in Y^X iff $(f_i x_i) \to fx$ in Y whenever $(x_i) \to x$ in X. We recall the following facts without proof.

Lemma 4.1.1.

1. *Any topological space with all topologically convergent sequences forms a limit space.*

2. *Any continuous map of topological spaces is continuous in the sense of limit spaces.*

In particular, the one-point compactification $\mathbb{N}_\infty$ and the Cantor space $\mathbf{2}^{\mathbb{N}}$ (together with their topologically convergent sequences) are limit spaces. The following is analogous to the Yoneda Lemma 3.4.2.

Lemma 4.1.2. *Convergent sequences in any limit space X are in one-to-one correspondence with the continuous maps $\mathbb{N}_\infty \to X$.*

4.2 The Kleene–Kreisel spaces as a full subcategory of C-spaces

In this section we present a *classical* proof that the full subcategory of simple objects in C-**Space** is isomorphic to the category of Kleene–Kreisel spaces. But, as discussed in Chapter 3, we have *constructive* proofs that this subcategory is cartesian closed, and has a natural numbers object, and also a fan functional without assuming Brouwerian axioms. Thus, this subcategory can be seen as a constructive, classically equivalent, substitute for the traditional manifestations of Kleene–Kreisel spaces. Notice that this section is the only part in the thesis in which we employ non-constructive arguments.

As recalled in the previous section, limit spaces provide an approach to the Kleene–Kreisel continuous functionals via sequence convergence [45]. Therefore, instead of Kleene's notion of countable functional or Kreisel's notion of continuous functional, we relate our C-spaces to limit spaces to show how the Kleene–Kreisel spaces can be calculated within our constructive model.

In this section, to avoid confusion, we reserve the terminologies *continuous function* for morphisms of topological spaces, *probe-continuous function* for morphisms of C-spaces, and *limit-continuous function* for morphisms of limit spaces.

We first prove the analogue of Lemma 4.1.1 for C-spaces.

Lemma 4.2.1.

1. *The continuous maps from the set $\mathbf{2}^{\mathbb{N}}$ with the usual Cantor topology to any topological space X form a* C*-topology on X.*

2. *Any continuous map of topological spaces is probe-continuous.*

Proof. We firstly show that the three probe axioms are satisfied.

(1) Clearly all constant maps are continuous.

(2) Let $p\colon \mathbf{2}^{\mathbb{N}} \to X$ be a continuous map and $t \in \mathrm{C}$. As t is uniformly continuous and thus continuous, the composite $p \circ t$ of two continuous maps is continuous.

(3) Let $p_0, p_1\colon \mathbf{2}^{\mathbb{N}} \to X$ be continuous maps. Then it is clear that the map $p\colon \mathbf{2}^{\mathbb{N}} \to X$ defined by $p(i\alpha) = p_i(\alpha)$ is also continuous.

Now let $f\colon X \to Y$ be a continuous map of topological spaces. Since any probe p is a continuous map $\mathbf{2}^{\mathbb{N}} \to X$, the composite $f \circ p$ is continuous and thus a probe on Y. Hence f is probe-continuous. □

In particular, $\mathbb{N}_\infty$ together with all continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}_\infty$ forms a C-space. Now we define functors between the categories **Lim** and C-**Space**. By the above lemmas, the following holds for any of the three notions of continuity considered in this section:

Lemma 4.2.2. *The following maps* $\mathfrak{r}$ *and* $\mathfrak{s}$ *are continuous, and* $\mathfrak{r}$ *is a retraction with section* $\mathfrak{s}$*:*

$$\begin{array}{rcl} \mathfrak{r}\colon \mathbf{2}^{\mathbb{N}} & \to & \mathbb{N}_\infty \\ 1^n 0\alpha & \mapsto & n \\ 1^\omega & \mapsto & \infty, \end{array} \qquad \begin{array}{rcl} \mathfrak{s}\colon \mathbb{N}_\infty & \to & \mathbf{2}^{\mathbb{N}} \\ n & \mapsto & 1^n 0^\omega \\ \infty & \mapsto & 1^\omega. \end{array}$$

For a limit space X, define the *limit probes* on X to be the limit continuous maps $\mathbf{2}^{\mathbb{N}} \to X$ w.r.t. the limit structure on $\mathbf{2}^{\mathbb{N}}$ given in Lemma 4.1.1. The following shows that limit spaces can be regarded as a full subcategory of C-spaces.

Lemma 4.2.3 (The functor $G\colon$ **Lim** $\to$ C-**Space**)**.**

1. *For any limit space X, the limit probes form a* C*-topology on X.*
2. *For any two limit spaces X and Y, a function $X \to Y$ is limit continuous if and only if it is continuous w.r.t. the limit probes.*

This gives a full and faithful functor $G\colon$ **Lim** $\to$ C-**Space** which on objects keeps the same underlying set but replaces the limit structure by the C-topology given by limit probes, and is the identity on morphisms.

Proof. (1) We need to show that the three probe axioms are satisfied.

(i) It is clear that any constant map is limit-continuous and thus a probe.

(ii) Let $p\colon \mathbf{2}^{\mathbb{N}} \to X$ be limit-continuous and $t \in \mathrm{C}$. Given any convergent sequence $(x_i) \to x_\infty$ in $\mathbf{2}^{\mathbb{N}}$, the induced map $x\colon \mathbb{N}_\infty \to \mathbf{2}^{\mathbb{N}}$ is continuous. Because t is uniformly continuous, the composite $t \circ x$ is continuous and thus a convergent sequence. Then we have $(p(t(x_i))) \to p(t(x_\infty))$ by the limit-continuity of p, and thus the composite $p \circ t$ is limit-continuous.

(iii) Given probes $p_0, p_1\colon \mathbf{2}^{\mathbb{N}} \to X$, i.e. p_0, p_1 are limit-continuous, we define a map $\bar{p}\colon \mathbf{2} \to (\mathbf{2}^{\mathbb{N}} \to X)$ by $\bar{p}(0) = p_0$ and $\bar{p}(1) = p_1$. By the discreteness of $\mathbf{2}$, this map is limit-continuous. Since both the head function, $h(\alpha) = \alpha_0$, and the tail function, $t(\alpha) = \lambda n.\alpha_{n+1}$, are limit-continuous, the map $p\colon \mathbf{2}^{\mathbb{N}} \to X$, defined by

$$p = \lambda\alpha.\bar{p}(h\alpha)(t\alpha),$$

is also limit-continuous, by the cartesian closedness of **Lim**. Clearly p is the unique amalgamation of p_0, p_1.

(2) Let X and Y be limit spaces. ($\Rightarrow$) Suppose that $f\colon X \to Y$ is a limit-continuous map. Then it is also probe-continuous w.r.t. the C-topologies given as above, because any probe p on X is a limit-continuous map, and the composite $f \circ p$ limit-continuous and hence a probe on Y. ($\Leftarrow$) Suppose that $f\colon X \to Y$ is a probe-continuous map, w.r.t. the C-topologies given as above. Given a convergent sequence $(x_i) \to x_\infty$, we know that the induced map $x\colon \mathbb{N}_\infty \to X$ is limit-continuous by Lemma 4.1.2. By (2($\Rightarrow$)), we know that x is probe-continuous, and hence so is the composite $f \circ x$. Since the retraction $\mathfrak{r}\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}_\infty$ is continuous and thus a probe on $\mathbb{N}_\infty$, the composite $f \circ x \circ \mathfrak{r}$ is a probe on Y, *i.e.* a limit-continuous map. As $(1^i0^\omega) \to 1^\omega$ in $\mathbf{2}^{\mathbb{N}}$, the sequence $(f(x(\mathfrak{r}(1^i0^\omega))))$ converges to $f(x(\mathfrak{r}(1^\omega)))$, which amounts to $(fx_i) \to fx_\infty$ by the definition of $\mathfrak{r}$. □

Lemma 4.2.4 (The functor $F\colon$ C-**Space** $\to$ **Lim**)**.**

1. *For any* C*-space X, the probe-continuous maps $\mathbb{N}_\infty \to X$ form a limit-structure on X.*

2. *For any two* C*-spaces X and Y, if a function $X \to Y$ is probe-continuous then it is limit-continuous w.r.t. the above limit-structures.*

This gives a functor $F\colon$ C-**Space** $\to$ **Lim** which on objects again keeps the same underlying set but replaces the C-topology by an appropriate limit structure, and is the identity on morphisms.

Proof. (1) We need to verify the three axioms of limit structure.

(i) Clearly any constant map $\mathbb{N}_\infty \to X$ is probe-continuous and thus a convergent sequence.

(ii) If $x\colon \mathbb{N}_\infty \to X$ is probe-continuous, *i.e.* $(x_i) \to x_\infty$, and (x_{fi}) is a subsequence of (x_i), then we extend the reindexing function $f\colon \mathbb{N} \to \mathbb{N}$ to $\tilde{f}\colon \mathbb{N}_\infty \to \mathbb{N}_\infty$ by defining $\tilde{f}(n) = f(n)$ for $n \in \mathbb{N}$ and $\tilde{f}(\infty) = \infty$. Once we show that $\tilde{f}$ is probe-continuous, then so is the composite $x \circ \tilde{f}$ and thus $(x_{fi}) \to x_\infty$. For endomaps $\mathbb{N}_\infty \to \mathbb{N}_\infty$, probe-continuity corresponds to continuity. Given an open set $U \subseteq \mathbb{N}_\infty$. If $\infty \notin U$ then U must be a finite subset of $\mathbb{N}$. Since f is injective and strictly increasing, the set $\tilde{f}^{-1}(U)$ is also a finite subset of $\mathbb{N}$ and thus open. If $\infty \in U$ then its complement $\bar{U}$ is a finite subset of $\mathbb{N}$. The complement $\overline{\tilde{f}^{-1}(U)} = \tilde{f}^{-1}(\bar{U})$ is a finite subset of $\mathbb{N}$ and thus $\tilde{f}^{-1}(U)$ is open.

(iii) Suppose that (x_i) is a sequence such that every subsequence of (x_i) has a subsequence converging to x_∞. We need to prove the the induce map $x\colon \mathbb{N}_\infty \to X$ is probe-continuous. Let p be a probe on $\mathbb{N}_\infty$. By the assumption, we have a subsequence (x_{fi}) which converges to x_∞. We extend the reindexing function f as above and get a continuous map $\bar{f}\colon \mathbb{N}_\infty \to \mathbb{N}_\infty$. We know that f is bijective and thus the inverse $\bar{f}^{-1}$ is also continuous, because any continuous bijection of compact Hausdorff spaces is a homeomorphism. Since $(x_{fi}) \to x_\infty$, the map $x \circ \bar{f}$ is probe-continuous, *i.e.* $x \circ \bar{f} \circ q$ is a probe for any probe q on $\mathbb{N}_\infty$. If we choose $q = \bar{f}^{-1} \circ p$ which is a probe by the continuity of $\bar{f}^{-1}$, then $x \circ \bar{f} \circ \bar{f}^{-1} \circ p = x \circ p$ is a probe on X.

(Notice that in both (ii) and (iii) we have used non-constructive arguments.)

(2) Let X and Y be C-spaces, and let $f\colon X \to Y$ be a probe-continuous map. Given a convergent sequence $(x_i) \to x_\infty$, *i.e.* a probe-continuous map $x\colon \mathbb{N}_\infty \to X$, the composite $f \circ x$ is also probe-continuous and hence a convergent sequence on Y. □

Lemma 4.2.5. *Limit spaces form a reflective subcategory of* C*-spaces.*

Proof. It remains to show that $F\colon$ C-**Space** $\to$ **Lim** is left adjoint to G, i.e. for any C-space X and limit space Y, we have $\mathbf{Lim}(FX, Y) \cong$ C-$\mathbf{Space}(X, GY)$ naturally. As the underlying sets remain the same when we apply the functors, this is equivalent to saying that a map $f\colon X \to Y$ is limit-continuous iff it is probe-continuous.

($\Rightarrow$) Suppose f is limit-continuous, i.e. if $x\colon \mathbb{N}_\infty \to X$ is probe-continuous then $(fx_i) \to fx_\infty$. Given a probe $p\colon \mathbf{2}^{\mathbb{N}} \to X$ on X, we want to show that $f \circ p$ is a probe on Y, i.e. $f \circ p$ is limit-continuous. Given a convergent sequence $\alpha\colon \mathbb{N}_\infty \to \mathbf{2}^{\mathbb{N}}$, i.e. α is probe-continuous, the composite $p \circ \alpha$ is also probe-continuous. Then by the limit-continuity of f we have that $(f(p\alpha^i))$ converges to $f(p\alpha^\infty)$.

($\Leftarrow$) Suppose f is probe-continuous. Given a probe-continuous function $x\colon \mathbb{N}_\infty \to X$ (a convergent sequence), we want to show that $(fx_i) \to fx_\infty$. Since the retraction $\mathfrak{r}\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}_\infty$ defined in Lemma 4.2.2 is continuous and thus probe-continuous, so is the composite $x \circ \mathfrak{r}$. By the probe-continuity of f, we have that $f \circ x \circ \mathfrak{r}$ is a probe on Y and thus limit-continuous. Since $(1^i 0^\omega) \to 1^\omega$ in $\mathbf{2}^{\mathbb{N}}$, the sequence $(f(x(\mathfrak{r}(1^i 0^\omega))))$ converges to $f(x(\mathfrak{r}(1^\omega)))$, which amounts to $(fx_i) \to fx_\infty$ by the definition of $\mathfrak{r}$. □

Lemma 4.2.6. *The reflector* $F\colon$ C-**Space** $\to$ **Lim** *preserves finite products.*

Proof. It is trivial to verify that terminal objects are preserved by F. Now we show that F preserves binary products. Let X and Y be C-spaces. Both $F(X \times_{\text{C-}\mathbf{Space}} Y)$ and $F(X) \times_{\mathbf{Lim}} F(Y)$ have the same underlying set, the cartesian products of X and Y. We need to show that their limit structures are the same. ($\Rightarrow$) Given a convergent sequence $z\colon \mathbb{N}_\infty \to X \times Y$ in $F(X \times_{\text{C-}\mathbf{Space}} Y)$, i.e. a probe-continuous map, clearly the pair $(z \circ \mathrm{pr}_1, z \circ \mathrm{pr}_2)$ is a convergent sequence in $F(X) \times_{\mathbf{Lim}} F(Y)$. ($\Leftarrow$) Given a convergent sequence (x, y) in $F(X) \times_{\mathbf{Lim}} F(Y)$, where both $x\colon \mathbb{N}_\infty \to X$ and $y\colon \mathbb{N}_\infty \to Y$ are probe-continuous, we define a map $z\colon \mathbb{N}_\infty \to X \times Y$ by $z_i = (x_i, y_i)$. Clearly z is probe-continuous and thus a convergent sequence in $F(X \times_{\text{C-}\mathbf{Space}} Y)$. One can easily see that, if a convergent sequence is transferred by one of the above directions and then by the other, it remains the same. □

In view of the above, we can regard **Lim** as a full subcategory of C-**Space**.

Lemma 4.2.7. *[49, Corollary A.1.5.9] Let $G\colon \mathcal{C} \to \mathcal{D}$ be a functor between cartesian closed categories, and suppose G has a left functor F. If G is full and faithful and F preserves binary products, then G is cartesian closed (i.e. G preserves finite products and exponentials).*

Lemma 4.2.8. *[49, Proposition A.4.3.1] Let $\mathcal{D}$ be a cartesian closed category, and $\mathcal{C}$ a reflective subcategory of $\mathcal{D}$, corresponding to a reflector F on $\mathcal{C}$. Then F preserves finite products iff (the class of objects of) $\mathcal{C}$ is an exponential ideal in $\mathcal{D}$ (i.e. the exponential C^D is in $\mathcal{C}$ whenever $C \in \mathcal{C}$ and $D \in \mathcal{D}$).*

The above two general categorical lemmas give the following:

Theorem 4.2.9.

1. *The functor* $G\colon$ **Lim** $\to$ C-**Space** *is cartesian closed.*

2. *Limit spaces form an exponential ideal of* C-**Space**.

Moreover, the discrete objects in these two categories coincide.

Lemma 4.2.10. *If* X *is a discrete* C*-space, then* $G(F(X)) = X$.

Proof. It suffices to prove that $p\colon \mathbf{2}^{\mathbb{N}} \to X$ is a probe on X iff it is a probe on $G(F(X))$.

($\Rightarrow$) This direction holds for any C-space X. Let p be a probe on X. Then p is a probe-continuous map by the Yoneda Lemma 3.4.2. We need to show that p is a probe on $G(F(X))$, *i.e.* p is limit-continuous. Given a convergent sequence $x\colon \mathbb{N}_\infty \to \mathbf{2}^{\mathbb{N}}$, we know that x is also a probe-continuous map. Thus the composite $p \circ x$ is probe-continuous and thus a convergent sequence on X.

($\Leftarrow$) Let p be a probe on $G(F(X))$. According to the definitions of G and F, this means that, for any continuous maps $x\colon \mathbb{N}_\infty \to \mathbf{2}^{\mathbb{N}}$ and $q\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}_\infty$, the composite $p \circ x \circ q$ is locally constant. We need to show that p is a probe on X, *i.e.* p is locally constant. For the sake of contradiction, assume that p fails to be locally constant. By classical logic, this amounts to

$$\forall m \in \mathbb{N}.\ \exists \alpha, \beta \in \mathbf{2}^{\mathbb{N}}.\ \alpha =_m \beta \wedge p\alpha \neq p\beta$$

which, by countable choice, defines two sequences $(\boldsymbol{\alpha}^i)$ and $(\boldsymbol{\beta}^i)$ such that (i) $\boldsymbol{\alpha}^m =_m \boldsymbol{\beta}^m$ and (ii) $p\boldsymbol{\alpha}^m \neq p\boldsymbol{\beta}^m$ for all $m \in \mathbb{N}$. Because of the compactness of $\mathbf{2}^{\mathbb{N}}$ and (i), there are subsequences $(\boldsymbol{\alpha}^{fi})$ and $(\boldsymbol{\beta}^{fi})$, both of which converge to the same point $\gamma \in \mathbf{2}^{\mathbb{N}}$ and satisfy $\boldsymbol{\alpha}^{fi} =_{fi} \boldsymbol{\beta}^{fi}$ for all $i \in \mathbb{N}$. Because the composites $\boldsymbol{\alpha} \circ f$ and $\boldsymbol{\beta} \circ f$, being convergent, are continuous, we know that both $p \circ (\boldsymbol{\alpha} \circ f) \circ \mathfrak{r}$ and $p \circ (\boldsymbol{\beta} \circ f) \circ \mathfrak{r}$ are locally constant, where $\mathfrak{r}$ is the retraction defined in Lemma 4.2.2. Let m to be the maximum of their moduli. By their local constancy and the fact $1^m 0^\omega =_m 1^\omega$, we have

$$p\boldsymbol{\alpha}^{fm} = p\left(\boldsymbol{\alpha}^{f(\mathfrak{r}(1^m 0^\omega))}\right) = p\left(\boldsymbol{\alpha}^{f(\mathfrak{r}(1^\omega))}\right) = p\gamma$$

$$p\boldsymbol{\beta}^{fm} = p\left(\boldsymbol{\beta}^{f(\mathfrak{r}(1^m 0^\omega))}\right) = p\left(\boldsymbol{\beta}^{f(\mathfrak{r}(1^\omega))}\right) = p\gamma$$

and thus $p\boldsymbol{\alpha}^{fm} = p\boldsymbol{\beta}^{fm}$ which contradicts (ii) and hence shows that p must be locally constant. □

Recall that Kleene-Kreisel spaces can be obtained within **Lim** by starting with the natural numbers object, and closed under finite products and function spaces. By Theorem 4.2.9 and Lemma 4.2.10, the full subcategory of C-**Space** generated by the same process is isomorphic to the above one of **Lim**, which leads to the following conclusion:

Theorem 4.2.11. *The Kleene–Kreisel spaces can be calculated within* C-**Space** *by starting from the natural numbers object and iterating products and exponentials.*

A more direct proof of this theorem is as follows: (0) $F(GX) = X$ for any limit space X. (1) If a map $p\colon \mathbf{2}^{\mathbb{N}} \to X$ is a probe on a C-space X, then it is also a probe on $G(FX)$. (2) If $X = G(FX)$ and $Y = G(FY)$, then $X \times Y = G(F(X \times Y))$. (3) If $Y = G(FY)$

then $Y^X = G(F(Y^X))$. The proofs of (0) and (1) are easy and those of (2) and (3) use (1). The advantage of the more abstract approach we have chosen is that it gives additional information.

4.3 The Kleene–Kreisel and full-type hierarchies

The *full type hierarchy* is the smallest full subcategory of **Set** containing the natural numbers and closed under exponentials. If we work in a constructive set theory (or type theory) with the Brouwerian axiom UC, it turns out that the full type hierarchy is equivalent to the Kleene–Kreisel hierarchy calculated within C-**Space**. It is interesting that other Brouwerian axioms such as more general forms of continuity or Bar Induction are not needed to prove the equivalence. In the proofs below, we explicitly assume UC whenever it is needed.

For a set X, one can take all maps $\mathbf{2}^{\mathbb{N}} \to X$ as probes on X. The resulting space is called *indiscrete*, and refer to the collection of all maps $\mathbf{2}^{\mathbb{N}} \to X$ as the *indiscrete* C-topology on X. It is clear that X is indiscrete iff for any C-space Y, all maps $Y \to X$ are continuous. This is equivalent to saying that the functor $\nabla\colon \mathbf{Set} \to \text{C-}\mathbf{Space}$ that endows a set with the indiscrete C-topology is right adjoint to the forgetful functor C-**Space** $\to$ **Set**. Moreover, the adjunction becomes an equivalence when restricted to indiscrete spaces:

Lemma 4.3.1. *The category of indiscrete* C*-spaces is equivalent to* **Set***.*

Lemma 4.3.2. *Indiscrete* C*-spaces form an exponential ideal.*

Proof. Let X be a C-space and Y an indiscrete C-space. Given $r\colon \mathbf{2}^{\mathbb{N}} \to Y^X$, for any $t \in \mathrm{C}$ and $p \in \mathrm{Probe}(X)$, the map $\lambda\alpha.r(t\alpha)(p\alpha)$ has codomain Y and thus a probe on Y. Therefore, all maps $\mathbf{2}^{\mathbb{N}} \to Y^X$ are probes on Y^X. □

The crucial, but easy, observation is this:

Lemma 4.3.3. *If UC holds in* **Set***, then the discrete space* $\mathbb{N}$ *is also indiscrete.*

Proof. By construction, the discrete topology consists of all *uniformly continuous* functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$. But if UC holds in **Set**, this amounts to *all* functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$, which, by definition, constitute the indiscrete topology, and hence $\mathbb{N}$ is indiscrete (and the discrete and indiscrete topologies are the same). □

The desired result follows directly from the previous lemmas:

Corollary 4.3.4. *If* UC *holds in* **Set***, then the full type hierarchy is equivalent to the Kleene–Kreisel hierarchy.*

This can be strengthened so that its converse also holds:

Theorem 4.3.5. *The forgetful functor from the Kleene–Kreisel hierarchy to the full type hierarchy is an equivalence if and only if UC holds in* **Set***.*

Proof. By the above lemmas, if UC holds in **Set**, then the adjunction restricts to an equivalence of the two hierarchies. Conversely, if it is an equivalence, then UC holds in **Set**, because, as we have seen, it always holds in C-**Space**. □

Two larger full subcategories of **Set** and C-**Space** are equivalent if UC holds.

Lemma 4.3.6.

1. *Finite products of indiscrete* C*-spaces are indiscrete.*
2. *If* UC *holds in the category of sets, then finite coproducts of indiscrete* C*-spaces are indiscrete.*

Proof. The first claim is trivial. If UC holds, then the space $\mathbf{2}$ is both discrete and indiscrete. If X and Y are indiscrete spaces, we construct a coproduct $X + Y$ as in the proof of Theorem 3.3.5. We also define, by cases, a map $i\colon X + Y \to \mathbf{2}$ which maps $\mathrm{in}_0\ x$ to 0 and $\mathrm{in}_1\ y$ to 1 for any $x \in X$ and $y \in Y$. As $\mathbf{2}$ is indiscrete, the map i is continuous. Given any map $r\colon \mathbf{2}^{\mathbb{N}} \to X + Y$, the composite $i \circ r$ is a probe on $\mathbf{2}$ and thus locally constant, i.e. there is a natural number n such that for all $s \in \mathbf{2}^n$ the composite $i \circ r \circ \mathrm{cons}_s$ is constant. If its value is 0, then $r \circ \mathrm{cons}_s$ maps all $\alpha \in \mathbf{2}^{\mathbb{N}}$ to $\mathrm{in}_0\ x$ for some $x \in X$, i.e. there is a map $p\colon \mathbf{2}^{\mathbb{N}} \to X$ such that $r(\mathrm{cons}_s\alpha) = \mathrm{in}_0(p\alpha)$; otherwise, there is a map $q\colon \mathbf{2}^{\mathbb{N}} \to Y$ such that $r(\mathrm{cons}_s\alpha) = \mathrm{in}_1(q\alpha)$. By the definition, the map r is a probe on $X + Y$. □

Define *extended hierarchies* by closing under finite products and coproducts, in addition to exponentials.

Theorem 4.3.7. *If* UC *holds in* **Set***, then the extended full type hierarchy is equivalent to the extended Kleene–Kreisel hierarchy.*

We also conjecture that, under the assumption that all set-theoretic maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous, the above equivalence can be extended to the dependent-type hierarchies in the categories **Set** and C-**Space**, in the sense that the structures of category with families [29] on **Set** and on C-**Space** are equivalent.

CHAPTER 5

Modelling simple types in C-spaces

Our C-spaces form a cartesian closed category with a natural numbers object, as proved in Chapter 3.3, which suffice to model Gödel's System T and its logic HA^ω. Moreover, there is a fan function in the category of C-spaces that continuously calculates moduli of uniform continuity (Theorem 3.5.2), which will be used to validate the uniform-continuity principle (UC) in both theories. In the case of system T, in which quantifiers are absent, we skolemize (UC) using (an additional constant for) the fan functional. We also recover a well known result, namely that a definable function $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ in the full type hierarchy is uniformly continuous, by establishing a logical relation between the full type hierarchy and the Kleene–Kreisel functionals. We emphasize that all the work in this Chapter has been implemented in intensional Martin-Löf type theory in Agda notation, as discussed in Chapters 7 and 8.

5.1 A continuous model of Gödel's System T

The *term language* of system T has a ground type N of natural numbers, binary product type $\times$ and function type $\to$. For our purposes, it is convenient (although not strictly necessary) to add a ground type $\mathsf{2}$ of booleans. The constants and equations associated to the ground types are

- the natural number $\mathsf{0}\colon \mathsf{N}$,
- the successor function $\mathsf{suc}\colon \mathsf{N} \to \mathsf{N}$, and
- the recursion combinator

$$\begin{array}{lcl} \mathsf{rec}\colon \sigma \to (\mathsf{N} \to \sigma \to \sigma) \to \mathsf{N} \to \sigma & & \\ \mathsf{rec}\ x\ f\ \mathsf{0} & = & x \\ \mathsf{rec}\ x\ f\ (\mathsf{suc}\ n) & = & f\ n\ (\mathsf{rec}\ x\ f\ n). \end{array}$$

- booleans $\mathsf{f}, \mathsf{t}\colon \mathsf{2}$,
- the case-distinction function:

$$\begin{array}{lcl} \mathsf{if}\colon \mathsf{2} \to \sigma \to \sigma \to \sigma & & \\ \mathsf{if}\ \mathsf{f}\ x\ y & = & x \\ \mathsf{if}\ \mathsf{t}\ x\ y & = & y, \end{array}$$

The atomic *formulas* in system T consist of equations between terms of the same type, and more complex *formulas* are obtained by combining these with the propositional connectives $\wedge$ and $\Rightarrow$ (the negation of a formula ϕ can of course be defined as $\phi \Rightarrow \mathsf{0} = \mathsf{suc}\ \mathsf{0}$).

The term language of system T can be interpreted in any cartesian closed category $\mathbf{C}$ with a natural numbers object $\mathbb{N}$ and a coproduct $\mathbf{2}$ (or $\mathbf{1}+\mathbf{1}$) of two copies of the terminal object [55]. Specifically, types are interpreted as objects: the ground type N is interpreted as the object $\mathbb{N}$, the ground type $\mathsf{2}$ as the coproduct $\mathbf{2}$, product types as products, and function types as exponentials. Contexts are interpreted inductively as products. And a term in context is interpreted as a morphism from the interpretation of its context to the one of its type. The constant rec and if are interpreted using the universal properties of $\mathbb{N}$ and $\mathbf{2}$ in the standard way [55].

Throughout this section (and the next), we use σ, τ to range over types, bold lower case letters $\boldsymbol{t}$, $\boldsymbol{u}$, $\boldsymbol{x}$, $\boldsymbol{f}$, $\boldsymbol{m}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ to range over terms, and ϕ, ψ to range over formulas.

Uniform continuity of T-definable functions. As we have seen, both the categories **Set** and C-**Space** are cartesian closed and have a natural numbers objects and a coproduct $\mathbf{1} + \mathbf{1}$, and hence serve as models. We now recover a well-known result, using a logical relation between these two models. In the following we use the semantic brackets $[\![-]\!]$ for the interpretation, and add **Set** and C-**Space** as subscripts to distinguish which model we are working with.

Definition 5.1.1. The *logical relation* R over the set-theoretical and C-space models is defined by

1. If σ is a T type, then $\mathrm{R}_\sigma \subseteq [\![\sigma]\!]_{\mathbf{Set}} \times [\![\sigma]\!]_{\text{C-}\mathbf{Space}}$ is defined by induction on type σ as follows:
   (a) $\mathrm{R}_\iota(a, a')$ iff $a = a'$, where ι is the ground type $\mathsf{2}$ or N;
   (b) $\mathrm{R}_{\sigma\to\tau}(f, f')$ iff, for any $a \in [\![\sigma]\!]_{\mathbf{Set}}$ and any $a' \in [\![\sigma]\!]_{\text{C-}\mathbf{Space}}$, if $\mathrm{R}_\sigma(a, a')$ then $\mathrm{R}_\tau(f(a), f'(a'))$.
2. If $\Gamma = x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n$ is a context, then $\mathrm{R}_\Gamma \subseteq [\![\Gamma]\!]_{\mathbf{Set}} \times [\![\Gamma]\!]_{\text{C-}\mathbf{Space}}$ is defined by $\mathrm{R}_\Gamma(\vec{a}, \vec{a'})$ iff $\mathrm{R}_{\sigma_i}(a_i, a'_i)$ for all $i \leq n$.
3. Given $f = [\![\Gamma \vdash \boldsymbol{t} : \tau]\!]_{\mathbf{Set}}$ and $f' = [\![\Gamma \vdash \boldsymbol{t} : \tau]\!]_{\text{C-}\mathbf{Space}}$, $\mathrm{R}(f, f')$ iff, for any $\vec{a} \in [\![\Gamma]\!]_{\mathbf{Set}}$ and any $\vec{a'} \in [\![\Gamma]\!]_{\text{C-}\mathbf{Space}}$, if $\mathrm{R}_\Gamma(\vec{a}, \vec{a'})$ then $\mathrm{R}_\tau(f(\vec{a}), f'(\vec{a'}))$.

With a proof by induction on types as usual, we can easily show that the interpretations of any T term in these two models are related.

Lemma 5.1.2. *If* $\Gamma \vdash \boldsymbol{t} : \tau$, *then* $\mathrm{R}([\![\Gamma \vdash \boldsymbol{t} : \tau]\!]_{\mathbf{Set}}, [\![\Gamma \vdash \boldsymbol{t} : \tau]\!]_{\text{C-}\mathbf{Space}})$.

We say that an element $x \in [\![\sigma]\!]_{\mathbf{Set}}$ in the set-theoretical model is T-*definable* if it is the interpretation of some closed T term, i.e. there exists a closed term $\boldsymbol{t} : \sigma$ such that $x = [\![\boldsymbol{t}]\!]_{\mathbf{Set}}$.

Theorem 5.1.3. *Any* T-*definable function* $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ *is uniformly continuous.*

Proof. If $f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ interprets the term $\boldsymbol{f} : (\mathsf{N}\to\mathsf{2})\to\mathsf{N}$, then f is related to the continuous map $[\![\boldsymbol{f}]\!]_{\text{C-}\mathbf{Space}}\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ according to the above lemma. By the definition of the logical relation, we can easily show that f is uniformly continuous. □

Validating the uniform-continuity principle in system T. The above shows that the *definable* functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ in the full type hierarchy are uniformly continuous, with uniform continuity formulated externally to the theory, in the model. We now show how to validate the internal principle of uniform continuity, working with C-spaces.

In a theory with quantifiers, such as HA^ω discussed in the next section, the principle UC is formulated as follows, as discussed in Chapter 2:

$$\vdash \forall \boldsymbol{f}\colon(\mathsf{N}\to 2)\to\mathsf{N}.\ \exists \boldsymbol{m}\colon\mathsf{N}.\ \forall \boldsymbol{\alpha},\boldsymbol{\beta}\colon\mathsf{N}\to 2.\ \boldsymbol{\alpha} =_{\boldsymbol{m}} \boldsymbol{\beta} \Rightarrow \boldsymbol{f}\boldsymbol{\alpha} = \boldsymbol{f}\boldsymbol{\beta}.$$

In order to express this in system T, which lacks quantifiers, we first treat $\Gamma \vdash \forall \boldsymbol{x}\colon\sigma.\phi$ as $\Gamma, \boldsymbol{x}\colon\sigma \vdash \phi$, and then we add a constant

$$\mathsf{fan}\colon ((\mathsf{N}\to 2)\to\mathsf{N})\to\mathsf{N}$$

to remove the existential quantifier by Skolemization, so that we get the purely equational formulation

$$\boldsymbol{f}\colon(\mathsf{N}\to 2)\to\mathsf{N}, \boldsymbol{\alpha}\colon\mathsf{N}\to 2, \boldsymbol{\beta}\colon\mathsf{N}\to 2 \ \vdash\ \boldsymbol{\alpha} =_{\mathsf{fan}(\boldsymbol{f})} \boldsymbol{\beta} \Rightarrow \boldsymbol{f}\boldsymbol{\alpha} = \boldsymbol{f}\boldsymbol{\beta}.$$

To formulate $\boldsymbol{\alpha} =_{\boldsymbol{m}} \boldsymbol{\beta}$, we define a term $\mathsf{agree}\colon (\mathsf{N}\to 2)\to(\mathsf{N}\to 2)\to\mathsf{N}\to 2$ by

$$\mathsf{agree}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ =\ \mathsf{rec}\ \mathsf{t}\ (\lambda \boldsymbol{i}.\ \lambda \boldsymbol{x}.\ \mathsf{min}\ (\mathsf{eq}\ \boldsymbol{\alpha}\boldsymbol{i}\ \boldsymbol{\beta}\boldsymbol{i})\ \boldsymbol{x}),$$

where $\mathsf{min}\colon 2\to 2\to 2$ gives the minimal boolean and $\mathsf{eq}\colon 2\to 2\to 2$ has value t iff its two arguments are the same, both of which can be defined using if. The idea is that

$$\mathsf{agree}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{m} = \mathsf{t} \ \text{ iff } \ \boldsymbol{\alpha} \text{ and } \boldsymbol{\beta} \text{ are equal up to the first } \boldsymbol{m} \text{ positions.}$$

Then the formula of UC that we are working with becomes

$$\boldsymbol{f}\colon(\mathsf{N}\to 2)\to\mathsf{N}, \boldsymbol{\alpha}\colon\mathsf{N}\to 2, \boldsymbol{\beta}\colon\mathsf{N}\to 2 \ \vdash\ \mathsf{agree}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ (\mathsf{fan}\ \boldsymbol{f}) = \mathsf{t} \Rightarrow \boldsymbol{f}\boldsymbol{\alpha} = \boldsymbol{f}\boldsymbol{\beta}.$$

We interpret types and terms of this theory in C-**Space** as before, while the meaning of the constant fan is given by the functional $\mathrm{fan}\colon \mathbb{N}^{\mathbf{2}^{\mathbb{N}}} \to \mathbb{N}$ constructed in Chapter 3.5. Here and in the next section, semantic brackets without explicit decorations refer to the C-**Space** interpretation. Formulas are interpreted inductively as follows. Given $\vec{\rho} \in [\![\Gamma]\!]$,

(1) $[\![\Gamma \vdash \boldsymbol{t} = \boldsymbol{u}]\!](\vec{\rho}) \ = [\![\Gamma \vdash \boldsymbol{t} : \sigma]\!](\vec{\rho}) = [\![\Gamma \vdash \boldsymbol{u} : \sigma]\!](\vec{\rho})$,

(2) $[\![\Gamma \vdash \phi \wedge \psi]\!](\vec{\rho}) \ = [\![\Gamma \vdash \phi]\!](\vec{\rho}) \times [\![\Gamma \vdash \psi]\!](\vec{\rho})$,

(3) $[\![\Gamma \vdash \phi \Rightarrow \psi]\!](\vec{\rho}) = [\![\Gamma \vdash \phi]\!](\vec{\rho}) \to [\![\Gamma \vdash \psi]\!](\vec{\rho})$,

where $=$ represents the identity type, $\times$ the binary product, and $\to$ the function space in the meta-theory. We say that C-**Space** *validates* $\Gamma \vdash \phi$ iff $[\![\Gamma \vdash \phi]\!](\vec{\rho})$ is inhabited for any $\vec{\rho} \in [\![\Gamma]\!]$.

Theorem 5.1.4. *The model of* C*-spaces validates* UC.

Proof. If the interpretation of the formula $\mathsf{agree}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{m} = \mathsf{t}$ is inhabited, then we have $[\![\boldsymbol{\alpha}]\!] =_{[\![\boldsymbol{m}]\!]} [\![\boldsymbol{\beta}]\!]$, with a proof by induction on $[\![\boldsymbol{m}]\!]$. In particular, the inhabitedness of the interpretation of $\mathsf{agree}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ (\mathsf{fan}\ \boldsymbol{f}) = \mathsf{t}$ implies $[\![\boldsymbol{\alpha}]\!] =_{\mathrm{fan}[\![\boldsymbol{f}]\!]} [\![\boldsymbol{\beta}]\!]$. According to the definition of fan, we have $[\![\boldsymbol{f}\boldsymbol{\alpha}]\!] = [\![\boldsymbol{f}\boldsymbol{\beta}]\!]$. □

5.2 A continuous realizability semantics of HA^ω

Heyting arithmetic over finite types, abbreviated HA^ω, is system T (with its equational theory and logical connectives) extended with the intuitionistic quantifiers [7, §V] [73, §3 and §9]. For technical convenience, we add a singleton type $\mathsf{1}$ to the inductive definition of types in T.

To any HA^ω formula ϕ we associate a T type $|\phi|$ of potential realizers. Then a *continuous realizer* of a formula $\Gamma \vdash \phi$ is a pair

$$(\vec{\rho}, e) \in [\![\, |\Gamma| \,]\!] \times [\![\phi]\!].$$

We call this a *continuous realizability semantics*.

Definition 5.2.1 (Continuous realizability)**.** The *types of potential realizers* of HA^ω formulas are given inductively as follows:

1. $|\boldsymbol{t} = \boldsymbol{u}| \;=\; \mathsf{1}$,
2. $|\phi \wedge \psi| \;=\; |\phi| \times |\psi|$,
3. $|\phi \Rightarrow \psi| \;=\; |\phi| \to |\psi|$,
4. $|\forall x{:}\sigma.\ \phi| \;=\; \sigma \to |\phi|$,
5. $|\exists x{:}\sigma.\ \phi| \;=\; \sigma \times |\phi|$.

Let Γ be a context and $\vec{\rho} \in [\![\Gamma]\!]$. The relation

$$(\vec{\rho}, e) \text{ realizes } \Gamma \vdash \phi$$

is defined by induction on formulas as follows:

1. $(\vec{\rho}, \star)$ realizes $\Gamma \vdash \boldsymbol{t} = \boldsymbol{u}$ iff $[\![\Gamma \vdash \boldsymbol{t} : \sigma]\!](\vec{\rho}) = [\![\Gamma \vdash \boldsymbol{u} : \sigma]\!](\vec{\rho})$, where $\star$ is the element of the terminal C-space,
2. $(\vec{\rho}, e)$ realizes $\Gamma \vdash \phi_0 \wedge \phi_1$ iff $(\vec{\rho}, \pi_i(e))$ realizes $\Gamma \vdash \phi_i$ for all $i \in \{0, 1\}$, where $e \in [\![\, |\phi_0| \,]\!] \times [\![\, |\phi_1| \,]\!]$,
3. $(\vec{\rho}, e)$ realizes $\Gamma \vdash \phi \Rightarrow \psi$ iff for all $a \in [\![\, |\phi| \,]\!]$ with $(\vec{\rho}, a)$ realizing $\Gamma \vdash \phi$, the pair $(\vec{\rho}, e(a))$ realizes $\Gamma \vdash \psi$, where $e\colon [\![\, |\phi| \,]\!] \to [\![\, |\psi| \,]\!]$ is continuous,
4. $(\vec{\rho}, e)$ realizes $\Gamma \vdash \forall \boldsymbol{x}{:}\sigma.\ \phi$ iff for all $a \in [\![\sigma]\!]$, the pair $((\vec{\rho}, a), e(a))$ realizes $\Gamma, \boldsymbol{x}{:}\sigma \vdash \phi$, where $e\colon [\![\sigma]\!] \to [\![\, |\phi| \,]\!]$ is continuous and $(\vec{\rho}, a) \in [\![\Gamma, x{:}\sigma]\!]$,
5. $(\vec{\rho}, e)$ realizes $\Gamma \vdash \exists \boldsymbol{x}{:}\sigma.\ \phi$ iff $((\vec{\rho}, \pi_0(e)), \pi_1(e))$ realizes $\Gamma, \boldsymbol{x}{:}\sigma \vdash \phi$, where $e \in [\![\sigma]\!] \times [\![\, |\phi| \,]\!]$.

We say a closed HA^ω formula ϕ is *realizable* if there exists $e \in [\![\, |\phi| \,]\!]$ such that $(\star, e)$ realizes $\vdash \phi$.

Theorem 5.2.2. *The principle* UC*, formulated as the following* HA^ω *formula*

$$\vdash \forall \boldsymbol{f}{:}(\mathsf{N}{\to}2){\to}\mathsf{N}.\ \exists \boldsymbol{m}{:}\mathsf{N}.\ \forall \boldsymbol{\alpha},\boldsymbol{\beta}{:}\mathsf{N}{\to}2.\ \mathsf{agree}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{m} = \mathsf{t} \Rightarrow \boldsymbol{f}(\boldsymbol{\alpha}) = \boldsymbol{f}(\boldsymbol{\beta})$$

is realized by the fan functional.

Proof. If $(\star, e)$ realizes UC, then e is a continuous map

$$\mathbb{N}^{\mathbf{2}^{\mathbb{N}}} \to \mathbb{N} \times (\mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}} \to \mathbf{1} \to \mathbf{1}).$$

By Definition 5.2.1, given any continuous $f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$, the pair $((\star, f), e(f))$ realizes

$$\boldsymbol{f}{:}(\mathsf{N}{\to}2){\to}\mathsf{N}\ \vdash\ \exists \boldsymbol{m}{:}\mathsf{N}.\ \forall \boldsymbol{\alpha},\boldsymbol{\beta}{:}\mathsf{N}{\to}2.\ \mathsf{agree}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{m} = \mathsf{t} \Rightarrow \boldsymbol{f}(\boldsymbol{\alpha}) = \boldsymbol{f}(\boldsymbol{\beta}).$$

If we define the first component of $e(f)$ to be $\mathrm{fan}(f)$, i.e. the modulus of uniform continuity of f, then we want that $((\star, f, \mathrm{fan}(f)), (\pi_1(e(f))))$ realizes

$$\boldsymbol{f}{:}(\mathsf{N}{\to}2){\to}\mathsf{N}, \boldsymbol{m}{:}\mathsf{N}\ \vdash\ \forall \boldsymbol{\alpha},\boldsymbol{\beta}{:}\mathsf{N}{\to}2.\ \mathsf{agree}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{m} = \mathsf{t} \Rightarrow \boldsymbol{f}(\boldsymbol{\alpha}) = \boldsymbol{f}(\boldsymbol{\beta}).$$

Now we write

$$\Gamma = \boldsymbol{f}{:}(\mathsf{N}{\to}2){\to}\mathsf{N}, \boldsymbol{m}{:}\mathsf{N}, \boldsymbol{\alpha}{:}\mathsf{N}{\to}2, \boldsymbol{\beta}{:}\mathsf{N}{\to}2.$$

Given any $\alpha, \beta \in \mathbf{2}^{\mathbb{N}}$, using the same argument as in the proof of Theorem 5.1.4, we can show that if $((\star, f, \mathrm{fan}(f), \alpha, \beta), \star)$ realizes $\Gamma \vdash \mathsf{agree}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{m} = \mathsf{t}$ then we have $\alpha =_{\mathrm{fan}(f)} \beta$. By the definition of fan, we have $f(\alpha) = f(\beta)$, i.e.

$$[\![\Gamma \vdash \boldsymbol{f}(\boldsymbol{\alpha})]\!](\star, f, \mathrm{fan}(f), \alpha, \beta) = [\![\Gamma \vdash \boldsymbol{f}(\boldsymbol{\beta})]\!](\star, f, \mathrm{fan}(f), \alpha, \beta).$$

Therefore, $((\star, f, \mathrm{fan}(f), \alpha, \beta), \star)$ realizes $\Gamma \vdash \boldsymbol{f}(\boldsymbol{\alpha}) = \boldsymbol{f}(\boldsymbol{\beta})$. □

CHAPTER 6

Modelling dependent types in sheaves

In the previous chapter, we modelled system T and HA^ω in the cartesian closed category of C-spaces, and then validated the uniform-continuity principle (UC) using the fan functional. In this chapter, we generalize this to a dependently typed theory. We begin with a short introduction to Martin-Löf type theory (MLTT) [60, 61, 69]. Then in Section 6.2, following Seely's method [66], we show that the Curry–Howard formulation of (UC) is validated in the locally cartesian closed category of C-spaces. It is well known that there is a coherence issue with substitutions arising in Seely's interpretation [25], which has been handled in a number of ways [25, 26, 29, 42]. We briefly recall the one via the notion of category with families (CwF) [29] in Section 6.3, and then construct CwF-structures in the categories of C-spaces (Section 6.4) and of sheaves (Section 6.5).

6.1 Martin-Löf type theory

Martin-Löf type theory (MLTT) provides an alternative foundation of constructive mathematics, based on the well-known slogan "propositions as types" [61]. The idea is that, in first order intuitionistic logic, propositions are interpreted as types, and proofs as terms, which is known as the *Curry–Howard interpretation*. Because of its computational content, MLTT can be regarded as a dependently typed programming language. The design of a number of proof assistants and programming languages is based on MLTT, including Coq [8] and Agda [11].

In this section, we recall the dependently typed theory based on Martin-Löf [61]. Our presentation is informal as in the first Chapter of the HoTT book [72], but is sufficient for the purpose of modelling the uniform-continuity principle (Section 6.2). We give another presentation of type theory in Section 6.3, via Cartmell's notion of generalized algebraic theory [14], to illustrate its close relation to the notion of category with families [29].

To present the theory, we use the following two forms of judgment:

1. $a : A$, which means that a is a *term* of *type* A, and
2. $a \equiv b$, which means that a and b are judgmentally equal objects.

We use the symbol $\equiv$ for judgmental equality, and reserve $=$ to denote identity types, as in the HoTT book [72]. Notice that a judgment may depend on a *context*, that is an ordered list of free variables occurring in the types or terms of that judgment. For the sake of readability, in this introductory section we omit contexts, but explicitly list the

free variables involved in the constructions of types and of terms. In Section 6.3 we deal with contexts explicitly.

MLTT has a more general version of function types, called a Π-*type*. The element of a Π-type is a *dependent function*, whose codomain type depends on the element of the domain type to which the function is applied. Given a type A and a family B which to each element $x : A$ assigns a type $B(x)$, the Π-type is written as

$$\Pi(x{:}A).B(x).$$

When B is a constant family, then the above is simply the ordinary function type $A \to B$. For a variable $x : A$, if we have a term $b : B(x)$, then we can construct a dependent function via λ-abstraction

$$\lambda x.b : \Pi(x{:}A).B(x).$$

If we apply a dependent function $f : \Pi(x{:}A).B(x)$ to a term $a : A$, then we get

$$f(a) : B(a).$$

Moreover, we have the following computation rules of Π-types: for any $b : B(x)$, $a : A$ and $f : \Pi(x{:}A).B(x)$,

$$(\lambda x.b)(a) \equiv b[a/x] \qquad f \equiv \lambda x.f(x)$$

where $b[a/x]$ means replacing all occurrences of x in b by a and hence is an element of $B(a)$.

MLTT also has a generalized version of product types, call a Σ-*type*. An element of a Σ-type is a *dependent pair* in which the type of the second component depends on the first component. Given a type A and a family B which to each element $x : A$ assigns a type $B(x)$, the Σ-type is written as

$$\Sigma(x{:}A).B(x).$$

When B is a constant family, then the above is simply the ordinary product type $A \times B$. By pairing terms $a : A$ and $b : B(a)$, we have

$$(a, b) : \Sigma(x{:}A).B(x).$$

The first projection from a Σ-type is a function

$$\mathrm{pr}_1 : (\Sigma(x{:}A).B(x)) \to A,$$

while the second one is dependent

$$\mathrm{pr}_2 : \Pi(w{:}\Sigma(x{:}A).B(x)).B(\mathrm{pr}_1(w)).$$

The computational rules of Σ-types include

$$\mathrm{pr}_1(a, b) \equiv a \qquad \mathrm{pr}_2(a, b) \equiv b \qquad w \equiv (\mathrm{pr}_1(w), \mathrm{pr}_2(w))$$

for any $a : A$, $b : B(a)$ and $w : \Sigma(x{:}A).B(x)$.

For any two elements of a type, MLTT has a type expressing their equality, called an

identity type. Given $a, b : A$, the identity type is written as

$$a =_A b.$$

An element of $a =_A b$ can be regarded as a proof that a and b are propositionally equal. To construct elements of an identity type, we have the following dependent function

$$\mathrm{refl} : \Pi(a : A).\ a =_A a$$

called *reflexivity*, which says that every element of type A is equal to itself. The induction principle for the identity type over elements of type A is given by the J-eliminator

$$\mathrm{J} : (\Pi(a{:}A).C(a, a, \mathrm{refl}_a)) \to \Pi(a{:}A).\Pi(b{:}A).\Pi(p{:}\ a =_A b).C(a, b, p)$$

for each family C which to elements $a, b : A$ and $p : a =_A b$ assigns a type $C(a, b, p)$. The computational rule of identity type is

$$\mathrm{J}(c, a, a, \mathrm{refl}_a) \equiv c(a)$$

for any $c : \Pi(a{:}A).C(a, a, \mathrm{refl}_a)$ and $a : A$. In other words, to prove a property for all elements $a, b : A$ and $p : a =_A b$, it suffices to consider all the cases where the elements are a, a and refl_a. We omit the indices and simply write $a = b$ and refl when they can be inferred from the context.

6.2 Modelling UC via the LCCC of C-spaces

In Chapter 5.1, we showed that C-spaces give a model of system T that validates the uniform-continuity principle (UC), namely

$$\forall f{:}\ \mathbf{2}^{\mathbb{N}} \to \mathbb{N}.\ \exists m \in \mathbb{N}.\ \forall \alpha, \beta \in \mathbf{2}^{\mathbb{N}}.\ \alpha =_m \beta \implies f\alpha = f\beta.$$

For this, we used the cartesian closedness of C-spaces to interpret simple types and formulas in system T, and we gave a skolemized version of (UC) using the fan functional in order to remove the quantifiers, which are absent from system T. In this section, we exploit the local cartesian closedness of C-spaces to model dependent types. In this case, the uniform-continuity principle is formulated as a closed type (CH-UC), via the Curry–Howard interpretation, rather than as a logical formula, namely

$$\Pi(f{:}\ (\mathsf{N} \to 2) \to \mathsf{N}).\ \Sigma(m{:}\mathsf{N}).\ \Pi(\alpha{:}\ \mathsf{N} \to 2).\ \Pi(\beta{:}\ \mathsf{N} \to 2).\ \alpha =_m \beta \to f\alpha = f\beta,$$

where 2 denotes the type of binary digits f and t, N denotes the type of natural numbers. Here $\alpha =_m \beta$ stands for $\Pi(i{:}\mathbb{N}).\ i < m \to \alpha_i = \beta_i$. For this, we have to introduce the less-than relation $<$ as a ground type, or equivalently define it as a Σ-type. Another way is to define a term $\mathsf{agree}{:}\ (\mathsf{N}{\to}2){\to}(\mathsf{N}{\to}2){\to}\mathsf{N}{\to}2$ using the primitive recursor as in Chapter 5.1. It is provable in type theory that

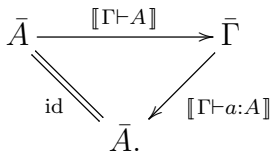
$$\mathsf{agree}\ \alpha\ \beta\ m = \mathsf{t} \quad \text{iff} \quad \Pi(i{:}\mathbb{N}).\ i < m \to \alpha_i = \beta_i.$$

Because all these definitions are equivalent, they would have equivalent interpretations in any model. Therefore, it does not matter which definition of $\alpha =_m \beta$ that we are working with. Our objective is to show that the type (CH-UC) is inhabited in the locally cartesian closed category of C-spaces.

As is well known, locally cartesian closed categories [66] and variations, such as categories with attributes [42] and categories with families [29], give models of dependent type theories. Seely's interpretation in locally cartesian closed categories [66] has a coherence issue with type substitution, as pointed out by Curien [25]. This problem can be addressed by changing the syntax to work with explicit substitutions [25], or by changing the semantics to work with categories with attributes [42] or categories with families [29]. A more recent discussion of categorical models of MLTT, relating Curien's [25] and Hofmann's [42] approaches can be found in [26].

Clairambault and Dybjer show that locally cartesian closed categories and categories with families are biequivalent [17]. Using one direction of this biequivalence, which amounts to Hofmann's construction [42], one can translate a locally cartesian closed category $\mathbf{C}$ to a category with families $(\mathbf{C}, T_{\mathbf{C}})$. Then one can show, by induction on types, that $\Gamma \vdash A$ is inhabited in $\mathbf{C}$ iff it is inhabited in $(\mathbf{C}, T_{\mathbf{C}})$. Therefore, we can ignore the coherence issue and work directly with the locally cartesian closed structure of C-**Space** to model the uniform-continuity principle.

Recall that, in Seely's model, a *type* in context $\Gamma \vdash A$ is interpreted as an object, i.e. a morphism $\bar{A} \to \bar{\Gamma}$, in the slice category $\mathbf{C}/\bar{\Gamma}$, and a *term* $\Gamma \vdash a : A$ is interpreted as a section of the interpretation of its type:



The right adjoint to the pullback functor interprets Π-types, its left adjoint interprets Σ-types, and equalizers interpret identity types. We say the model $\mathbf{C}$ *validates* $\Gamma \vdash A$ iff the interpretation $[\![\Gamma \vdash A]\!]$ has sections.

Theorem 6.2.1. *The locally cartesian closed category of* C*-spaces validates the Curry–Howard formulation* (CH-UC) *of the uniform-continuity principle.*

Proof. It is enough to show that the domain of $[\![\vdash \text{CH-UC}]\!]$ is inhabited. The proof is essentially the same as that of Theorem 5.1.4, which is carried out using the fan functional.

Since the space $\mathrm{dom}([\![\vdash (\mathsf{N} \to 2) \to \mathsf{N}]\!])$ is equivalent to the exponential $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$ in C-**Space**, the underlying set of $\mathrm{dom}([\![\vdash \text{CH-UC}]\!])$ is equivalent to the set of continuous functions $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}} \to \mathrm{dom}(u)$, where u is the interpretation of

$$f : (\mathsf{N} \to 2) \to \mathsf{N} \vdash \Sigma(m{:}\mathsf{N}).\ \Pi(\alpha : \mathsf{N} \to 2).\ \Pi(\beta : \mathsf{N} \to 2).\ \alpha =_m \beta \to f\alpha = f\beta.$$

The underlying set of $\mathrm{dom}(u)$ is equivalent to the set of pairs (f, m), where $f \in \mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$ and $m \in \mathbb{N}$, such that $f\alpha = f\beta$ whenever $\alpha =_m \beta$. By the definition of the fan functional, the pair $(f, \mathrm{fan}(f))$ is clearly in $\mathrm{dom}(u)$. Therefore, we have a map

$$(f \mapsto (f, \mathrm{fan}(f)) : \mathbb{N}^{\mathbf{2}^{\mathbb{N}}} \to \mathrm{dom}(u)$$

which is continuous because $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$ is discrete by Lemma 3.5.1. □

Notice that the space $\mathrm{dom}(u)$ in the above proof consists of tuples (f, m, ϕ) which satisfy some certain conditions. These conditions, together with the continuous map ϕ, amount to saying that m is a modulus of uniform continuity of the map f. Notice that this holds in various flavours of (extensional and intensional) MLTT, and that system T can be regarded as a subsystem of MLTT, and that the proof given here is essentially the same as the one given above for system T, in a slightly different language.

6.3 Categories with families

In this section we define the notion of model of MLTT as a *category with families* (CwF), and take the syntax of MLTT to be the initial such model, following Dybjer [29, 17]. To give such a model, we start with a base category $\mathbf{C}$ whose objects are referred to as contexts, and whose morphisms are referred to as substitutions, and we assume a terminal context, written []. We need to impose additional structure to the category $\mathbf{C}$ in order to be able to model types in context and terms of types in context. This is taken to be a functor from $\mathbf{C}$ to the category of families of sets. A so called context comprehension operation is also imposed to $\mathbf{C}$ which extends a given context with a type in that context. Moreover, other extra structures may be added to $\mathbf{C}$ in order to model certain types (*e.g.* natural numbers) or type formers (*e.g.* Π- and Σ-types) of a dependently typed theory.

The notion of category with families can be described as a *generalized algebraic theory* (GAT) in the sense of Cartmell [14]. Such a theory contains sorts, operators and equations of well-formed objects. There are four sorts, corresponding to the following four forms of sequent:

(1) "$\Gamma \vdash$" means that Γ is a *context*.

(2) "$\sigma \colon \Delta \to \Gamma$" means that σ is a *substitution*.

(3) "$\Gamma \vdash A$" means that A is a *type* in context Γ.

(4) "$\Gamma \vdash u \colon A$" means that u is a *term* of type A in context Γ.

Operators are introduced via typing rules for forming contexts (*e.g.* context extension), substitutions (*e.g.* substitution composition), types (*e.g.* Π- and Σ-type formers) and terms (*e.g.* associated constructors).

In this section, we also recall the GAT-presentation of a CwF-structure immediately after giving the corresponding categorical definitions. For both presentations, we use Γ, Δ, Θ to range over contexts, σ, τ, ν over substitutions, A, B over types, and u, v, w over terms.

Base category of contexts and substitutions. A category with families contains a *base category* $\mathbf{C}$ whose objects represent *contexts* and whose morphisms represent *substitutions*. In the GAT-presentation, this corresponds to the following typing rules

$$\frac{\Gamma \vdash}{1 \colon \Gamma \to \Gamma} \qquad \frac{\sigma \colon \Delta \to \Gamma \qquad \tau \colon \Theta \to \Delta}{\sigma \circ \tau \colon \Theta \to \Gamma}$$

with the following equations

$$1 \circ \sigma = \sigma = \sigma \circ 1 \qquad (\sigma \circ \tau) \circ \nu = \sigma \circ (\tau \circ \nu).$$

The intuition of an identity substitution is to replace the free variables in an expression by those free variables themselves. And composition of substitutions can be regarded as performing substitutions one by one. The equations of types and of terms below will make this clearer.

Moreover, the category $\mathbf{C}$ has a *terminal* object $[]$ representing the *empty context*. The corresponding GAT-description is given by

$$\frac{}{[] \vdash} \qquad \frac{\Gamma \vdash}{\langle\rangle : \Gamma \to []}$$

with equations

$$\langle\rangle \circ \sigma = \langle\rangle \qquad 1_{[]} = \langle\rangle.$$

Here $\langle\rangle$ is the unique substitution to the terminal context which expresses the weakening rule for closed expressions, *i.e.* if an expression is derivable in the empty context then it can be derived from any context. The two equations guarantee the uniqueness of $\langle\rangle$, as for any substitution $\sigma \colon \Gamma \to []$ it holds that $\sigma = 1_{[]} \circ \sigma = \langle\rangle \circ \sigma = \langle\rangle$.

Types, terms and their substitutions. Let **Fam** be the category of families of sets. Recall that an object in **Fam** is a family of sets $\{A_i\}_{i \in I}$ and that a morphism $\{A_i\}_{i \in I} \to \{B_j\}_{j \in J}$ consists of a function $f \colon I \to J$ and a family of functions $g_i \colon A_i \to B_{f(i)}$ indexed by $i \in I$. A category with families also contains a functor $T \colon \mathbf{C}^{\mathrm{op}} \to \mathbf{Fam}$. It maps each context $\Gamma \in \mathbf{C}$ to a family of sets $\{\mathrm{Term}(\Gamma, A)\}_{A \in \mathrm{Type}(\Gamma)}$, where

- $\mathrm{Type}(\Gamma)$ represents the set of *types* in Γ, and
- $\mathrm{Term}(\Gamma, A)$ represents the set of *terms* of type A in context Γ.

For each substitution $\sigma \colon \Delta \to \Gamma$, the **Fam**-morphism $T(\sigma) \colon T(\Gamma) \to T(\Delta)$ consists of

- a *type substitution* function mapping $A \in \mathrm{Type}(\Gamma)$ to $A[\sigma] \in \mathrm{Type}(\Delta)$, and
- a family of *term substitution* functions in which, for each $A \in \Gamma$, there is a function mapping $u \in \mathrm{Term}(\Gamma, A)$ to $u[\sigma] \in \mathrm{Term}(\Delta, A[\sigma])$.

The GAT-presentation of the functor T is the following:

$$\frac{\Gamma \vdash A \qquad \sigma \colon \Delta \to \Gamma}{\Delta \vdash A[\sigma]} \qquad \frac{\Gamma \vdash u : A \qquad \sigma \colon \Delta \to \Gamma}{\Delta \vdash u[\sigma] : A[\sigma]}$$

with equations

$$A[1] = A \qquad A[\sigma][\tau] = A[\sigma \circ \tau] \qquad u[1] = u \qquad u[\sigma][\tau] = u[\sigma \circ \tau].$$

The intuition of the two equations with identity substitution is that by replacing the free variables in a type (or a term) by themselves one of course gets the same type (or term). And the other two equations express the idea that the composite of two substitutions is equivalent to perform them one by one.

Context comprehension. In addition to the base category $\mathbf{C}$ and the function T, a category with families also contains an operation of *context comprehension*. To each context $\Gamma \in \mathbf{C}$ and type $A \in \mathrm{Type}(\Gamma)$, it associates a context $\Gamma.A$, a substitution $\mathrm{p}\colon \Gamma.A \to \Gamma$, and a term $\mathrm{q} \in \mathrm{Term}(\Gamma.A, A[\mathrm{p}])$, satisfying the following universal property: for any context $\Delta \in \mathbf{C}$, substitution $\sigma\colon \Delta \to \Gamma$ and term $u \in \mathrm{Term}(\Delta, A[\sigma])$, there is a *unique* substitution $(\sigma, u)\colon \Delta \to \Gamma.A$ such that $\mathrm{p} \circ (\sigma, u) = \sigma$ and $\mathrm{q}[(\sigma, u)] = u$. The GAT-presentation of the context comprehension operation is the following:

$$\frac{\Gamma \vdash A}{\Gamma.A \vdash} \qquad \frac{\Gamma \vdash A}{\mathrm{p}\colon \Gamma.A \to \Gamma} \qquad \frac{\Gamma \vdash A}{\Gamma.A \vdash \mathrm{q} : A[\mathrm{p}]}$$

$$\frac{\sigma\colon \Delta \to \Gamma \qquad \Gamma \vdash A \qquad \Delta \vdash u : A[\sigma]}{(\sigma, u)\colon \Delta \to \Gamma.A}$$

with equations

$$\mathrm{p} \circ (\sigma, u) = \sigma \qquad \mathrm{q}[(\sigma, u)] = u \qquad (\mathrm{p}, \mathrm{q}) = 1 \qquad (\sigma, u) \circ \delta = (\sigma \circ \delta, u[\delta]).$$

Notice that in the categorical definition the last two equations can be derived from the uniqueness property of substitution extension.

It may be helpful to think of a context as a (dependent) sequence of types, and a substitution as a (dependent) sequence of terms. The extended context $\Gamma.A$ is obtained by adding A to the end of the sequence Γ. The substitution $\mathrm{p}\colon \Gamma.A \to \Gamma$ expresses the weakening rule, *i.e.* from an expression in a context Γ we can get one in the extended context $\Gamma, x{:}A$. The term $\Gamma.A \vdash \mathrm{q} : A[\mathrm{p}]$ gives the last variable in the context $\Gamma.A$. And the extended substitution (σ, u) is obtained by adding u to the end of the sequence σ.

An example of a category with families. Up to this point, we have presented the full definition of category with families which gives a model of the minimal fragment of MLTT. In summary, a category with families consists of a base category $\mathbf{C}$, a functor $T\colon \mathbf{C}^{\mathrm{op}} \to \mathbf{Fam}$ and a context comprehension operation. A basic example is obtained by choosing $\mathbf{C}$ to be the category **Set** of sets and functions and defining the following:

- The terminal object $[]$ of **Set** is a singleton set.
- $\mathrm{Type}(\Gamma)$ is the set of Γ-indexed small sets, i.e. if $A \in \mathrm{Type}(\Gamma)$ then $A = \{A_\gamma\}_{\gamma \in \Gamma}$ where each A_γ is a small set.
- $\mathrm{Term}(\Gamma, A)$ is the dependent product $\prod_{\gamma \in \Gamma} A_\gamma$, i.e. elements in $\mathrm{Term}(\Gamma, A)$ are dependent functions.
- Given $A \in \mathrm{Type}(\Gamma)$ and $\sigma\colon \Delta \to \Gamma$, we define $A[\sigma] \in \mathrm{Type}(\Delta)$ by $(A[\sigma])_\delta = A_{\sigma(\delta)}$ for each $\delta \in \Delta$.
- Given $t \in \mathrm{Term}(\Gamma, A)$ and $\sigma\colon \Delta \to \Gamma$, we define $t[\sigma] \in \mathrm{Term}(\Delta, A[\sigma])$ by $(t[\sigma])(\delta) = t(\sigma(\delta))$ for each $\delta \in \Delta$.
- Given $\Gamma \in \mathbf{Set}$ and $A \in \mathrm{Type}(\Gamma)$, we define $\Gamma.A$ to be the dependent sum $\sum_{\gamma \in \Gamma} A_\gamma$. Then p and q are the first and second projections, and substitution extension is defined using the universal property of dependent sums.

Moreover, the category **Set** admits additional structures, such as those introduced below, to interpret different types. All these structures on **Set** can be easily implemented in intensional MLTT.

Σ-types. A category with families supports Σ-*types* if and only if

- for any types $A \in \mathrm{Type}(\Gamma)$ and $B \in \mathrm{Type}(\Gamma.A)$ we have a type $\Sigma AB \in \mathrm{Type}(\Gamma)$,
- for any elements $u \in \mathrm{Term}(\Gamma, A)$ and $v \in \mathrm{Term}(\Gamma, B[(1,u)])$ we have an element $(u,v) \in \mathrm{Term}(\Gamma, \Sigma AB)$, and
- for any element $w \in \mathrm{Term}(\Gamma, \Sigma AB)$ we have elements $\mathrm{pr}_1(w) \in \mathrm{Term}(\Gamma, A)$ and $\mathrm{pr}_2(w) \in \mathrm{Term}(\Gamma, B[(1, \mathrm{pr}_1(w))])$,

such that the following equations hold:

$$(\mathrm{pr}_1(w), \mathrm{pr}_2(w)) = w \qquad \mathrm{pr}_1(u,v) = u \qquad \mathrm{pr}_2(u,v) = v$$

$$(\Sigma AB)[\sigma] = \Sigma(A[\sigma])(B[(\sigma \circ \mathrm{p}, \mathrm{q})]) \qquad (u,v)[\sigma] = (u[\sigma], v[\sigma])$$
$$\mathrm{pr}_1(w)[\sigma] = \mathrm{pr}_1(w[\sigma]) \qquad \mathrm{pr}_2(w)[\sigma] = \mathrm{pr}_2(w[\sigma]).$$

The corresponding GAT-presentation is given by the following typing rules:

$$\frac{\Gamma \vdash A \qquad \Gamma.A \vdash B}{\Gamma \vdash \Sigma AB} \qquad \frac{\Gamma \vdash u : A \qquad \Gamma \vdash v : B[(1,u)]}{\Gamma \vdash (u,v) : \Sigma AB}$$
$$\frac{\Gamma \vdash w : \Sigma AB}{\Gamma \vdash \mathrm{pr}_1(w) : A} \qquad \frac{\Gamma \vdash w : \Sigma AB}{\Gamma \vdash \mathrm{pr}_2(w) : B[(1, \mathrm{pr}_1(w))]}$$

together with the same equations as above.

Π-types. A category with families supports Π-*types* if and only if

- for any types $A \in \mathrm{Type}(\Gamma)$ and $B \in \mathrm{Type}(\Gamma.A)$ we have $\Pi AB \in \mathrm{Type}(\Gamma)$,
- for any element $t \in \mathrm{Term}(\Gamma.A, B)$ we have an element $\lambda t \in \mathrm{Term}(\Gamma, \Pi AB)$, and
- for any elements $w \in \mathrm{Term}(\Gamma, \Pi AB)$ and $u \in \mathrm{Term}(\Gamma, A)$ we have an element $\mathrm{app}(w,u) \in \mathrm{Term}(\Gamma, B[(1,u)])$,

such that the following equations hold:

$$\mathrm{app}(\lambda v, u) = v[(1,u)] \qquad \lambda(\mathrm{app}(w[\mathrm{p}], \mathrm{q})) = w$$

$$(\Pi AB)[\sigma] = \Pi(A[\sigma])(B[(\sigma \circ \mathrm{p}, \mathrm{q})])$$
$$(\lambda v)[\sigma] = \lambda(v[(\sigma \circ \mathrm{p}, \mathrm{q})]) \qquad \mathrm{app}(w,u)[\sigma] = \mathrm{app}(w[\sigma], u[\sigma]).$$

The corresponding GAT-presentation is given by the following typing rules:

$$\frac{\Gamma \vdash A \qquad \Gamma.A \vdash B}{\Gamma \vdash \Pi AB} \qquad \frac{\Gamma.A \vdash v : B}{\Gamma \vdash \lambda v : \Pi AB} \qquad \frac{\Gamma \vdash w : \Pi AB \qquad \Gamma \vdash u : A}{\Gamma \vdash \mathrm{app}(w,u) : B[(1,u)]}$$

together with the same equations as above.

The construction of category with families and the additional structures to support Σ- and of Π-types are well-known [29, 21]. Now we introduce the structure for intensional identity types. We adopt the basic construction of [16], and propose the equations with substitutions that the type former Id and the eliminator J have to satisfy.

Intensional identity types. A category with families supports *intensional identity types* if and only if

- for any type $A \in \mathrm{Type}(\Gamma)$ we have a type $\mathrm{Id}_A \in \mathrm{Type}(\Gamma.A.A[\mathrm{p}])$,
- for any type $A \in \mathrm{Type}(\Gamma)$ we have a substitution $\mathrm{R}\colon \Gamma.A \to \Gamma.A.A[\mathrm{p}].\mathrm{Id}_A$, and
- for any type $B \in \mathrm{Type}(\Gamma.A.A[\mathrm{p}].\mathrm{Id}_A)$ and term $u \in \mathrm{Term}(\Gamma.A, B[\mathrm{R}])$ we have a term $\mathrm{J}(u) \in \mathrm{Term}(\Gamma.A.A[\mathrm{p}].\mathrm{Id}_A, B)$,

such that the following equations hold:

$$\mathrm{p} \circ \mathrm{R} = (1, \mathrm{q}) \qquad (\mathrm{J}(u))[\mathrm{R}] = u$$

$$\mathrm{Id}_A[((\sigma \circ \mathrm{p}, \mathrm{q}) \circ \mathrm{p}, \mathrm{q})] = \mathrm{Id}_{A[\sigma]} \qquad \mathrm{J}(u)[(((\sigma \circ \mathrm{p}, \mathrm{q}) \circ \mathrm{p}, \mathrm{q}) \circ \mathrm{p}, \mathrm{q})] = \mathrm{J}(u[(\sigma \circ \mathrm{p}, \mathrm{q})])$$

The corresponding GAT-presentation is given by the following typing rules:

$$\frac{\Gamma \vdash A}{\Gamma.A.A[\mathrm{p}] \vdash \mathrm{Id}_A} \qquad \frac{\Gamma \vdash A}{\mathrm{R}\colon \Gamma.A \to \Gamma.A.A[\mathrm{p}].\mathrm{Id}_A}$$

$$\frac{\Gamma.A.A[\mathrm{p}].\mathrm{Id}_A \vdash B \qquad \Gamma.A \vdash u\colon B[\mathrm{R}]}{\Gamma.A.A[\mathrm{p}].\mathrm{Id}_A \vdash \mathrm{J}(u)\colon B}$$

together with the same equations as above.

Intuitively $\mathrm{Id}_A \in \mathrm{Type}(\Gamma.A.A[\mathrm{p}])$ is the type that the last two variables in the context $\Gamma.A.A[\mathrm{p}]$ are equal. In another presentation of MLTT in Section 6.1, we have a constructor refl, which is a proof that any element is equal to itself, to introduce elements of an identity type. Here, the morphism $\mathrm{R}\colon \Gamma.A \to \Gamma.A.A[\mathrm{p}].\mathrm{Id}_A$ is playing the role of refl as intuitively R replaces the occurrences of variables $a, b : A$ and $p : a =_A b$ in an expression by a, a and refl_a. In fact the element refl can be defined as the term $\mathrm{q}[\mathrm{R}]$ of type $\mathrm{Id}_A[(1, \mathrm{q})]$ in context $\Gamma.A$ which does expresses that the last variable in $\Gamma.A$ is equal to itself. One can of course equivalently add refl as a constant and then define the morphism R using q and refl. But we find R to be more convenient to work with, *e.g.* to formulate the J-eliminator and its computational rule.

6.4 A continuous model of dependent types

In the Section 6.2, we employ the locally cartesian closed category of C-spaces to model Martin-Löf type theory, following Seely's method [66]. To avoid the coherence issue of this interpretation [25], we also work with the notion of category with families [29] which is recalled in Section 6.3. Instead of applying Hofmann's construction [42] to get a CwF-structure from a locally cartesian closed category, we directly develop one on the category of C-spaces. The idea of this model is that each context is (interpreted as) a C-space, while a type on a context Γ consists of a family of sets indexed by the underlying set of Γ and a family of C-topologies indexed by the probes on Γ.

The meta-theory for developing the CwF-structure on C-spaces in this section is some form of constructive set theory. It is an open problem whether this can be developed in intensional MLTT possibly extended with the axiom of function extensionality.

In this section, we write Γ, Δ, Θ to denote C-spaces, as they represent contexts, σ, τ, ν to denote continuous maps, as they represent substitutions, A, B to denote (interpretations of) types, and u, v, w to denote (interpretations of) terms.

The base category. Recall that a C-*space* is a set Γ equipped with a C-*topology* P consisting of maps $\mathbf{2}^{\mathbb{N}} \to \Gamma$, called *probes* on Γ, satisfying the following conditions:

(s1) For all $\gamma \in \Gamma$, the map $\lambda\alpha.\gamma$ is in P.

(s2) If $p \in P$ and $t \in \mathrm{C}$, then $p \circ t \in P$.

(s3) For any $p_0, p_1 \in P$, the unique map $p\colon \mathbf{2}^{\mathbb{N}} \to \Gamma$ defined by $p(i\alpha) = p_i(\alpha)$ is in P.

The above three conditions are called the *probe axioms*. Notice that the condition (s3) is logically equivalent to

(s3') For any $n \in \mathbb{N}$ and $\{p_s \in P\}_{s \in \mathbf{2}^n}$, the unique map $p\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ defined by $p(s\alpha) = p_s(\alpha)$ is in P.

A *continuous* map of C-spaces (Γ, P) and (Δ, Q) is a map $\sigma\colon \Gamma \to \Delta$ such that $\sigma \circ p \in Q$ whenever $p \in P$.

We abbreviate Γ for the C-space $(|\Gamma|, \mathrm{Probe}(\Gamma))$, where $|\Gamma|$ stands for the underlying set and $\mathrm{Probe}(\Gamma)$ for the C-topology on $|\Gamma|$, and often write Γ to mean $|\Gamma|$.

For any C-space Γ, clearly the identity map $1\colon \Gamma \to \Gamma$ is continuous. It is easy to verify that composition preserves continuity of maps. Therefore, C-spaces form a category, which is written as C-**Space**. This category has a *terminal object*, which is a singleton set $\mathbf{1} = \{\star\}$ equipped with the unique map $\mathbf{2}^{\mathbb{N}} \to \mathbf{1}$ as the only probe.

Types and terms. Given a C-space Γ, a *type* $\Gamma \vdash A$ is family $\{A_\gamma\}_{\gamma \in \Gamma}$ of sets equipped with a family $\{Q_p\}_{p \in \mathrm{Probe}(\Gamma)}$ of collections Q_p consisting of dependent maps $\Pi_{\alpha:\mathbf{2}^{\mathbb{N}}} A_{p(\alpha)}$, also called *probes*, satisfying the following conditions:

(t1) For all $\gamma \in \Gamma$ and $a \in A_\gamma$, the map $\lambda\alpha.a$ is in $Q_{\lambda\alpha.\gamma}$.

(t2) If $p \in \mathrm{Probe}(\Gamma)$, $q \in Q_p$ and $t \in \mathrm{C}$, then $q \circ t \in Q_{(p \circ t)}$.

(t3) For any $p \in \mathrm{Probe}(\Gamma)$, $q_0 \in Q_{p \circ \mathrm{cons}_0}$ and $q_1 \in Q_{p \circ \mathrm{cons}_1}$, the unique map $q\colon \Pi_{\alpha:\mathbf{2}^{\mathbb{N}}} A_{p(\alpha)}$ defined by $q(i\alpha) = q_i(\alpha)$ is in Q_p.

The following condition is logically equivalent to (t3):

(t3') For any $p \in \mathrm{Probe}(\Gamma)$, $n \in \mathbb{N}$ and $\{q_s \in Q_{p \circ \mathrm{cons}_s}\}_{s \in \mathbf{2}^n}$, the unique map $q\colon \Pi_{\alpha:\mathbf{2}^{\mathbb{N}}} A_{p(\alpha)}$ defined by $q(s\alpha) = q_s(\alpha)$ is in Q_p.

We also abbreviate A for a type $(|A|, \mathrm{Probe}(A))$, where $|A|$ stands for the underlying family and $\mathrm{Probe}(A)$ for the family of collections of dependent probes, and often write A to mean $|A|$.

Given a type $\Gamma \vdash A$, a *term* $\Gamma \vdash u : A$ is a dependent function $u\colon \Pi_{\gamma:\Gamma} A_\gamma$ such that $u \circ p \in \mathrm{Probe}(A)_p$ whenever $p \in \mathrm{Probe}(\Gamma)$.

Substitutions of types and terms. Given a type $\Gamma \vdash A$ and a continuous map $\sigma\colon \Delta \to \Gamma$, we define the *substituted type* $\Delta \vdash A[\sigma]$ by

$$A[\sigma]_\delta \;:\equiv\; A_{\sigma(\delta)}$$

for $\delta \in \Delta$. Given a probe $p \in \mathrm{Probe}(\Delta)$, we define

$$\mathrm{Probe}(A[\sigma])_p \;:\equiv\; \mathrm{Probe}(A)_{(\sigma\circ p)}.$$

It is well-defined due to the continuity of σ. The three conditions of types are easy to verify. But in a type-theoretic treatment this requires a closer look due to the presence of proof relevance, which will be discussed further in Chapter 7.3.

Given a term $\Gamma \vdash u : A$ and a continuous map $\sigma\colon \Delta \to \Gamma$, we define the *substituted term* $\Delta \vdash u[\sigma] : A[\sigma]$ by

$$u[\sigma](\delta) \;:\equiv\; u(\sigma(\delta))$$

for $\delta \in \Delta$. It is easy to check that it satisfies the condition of terms.

Since substitutions of types and terms are defined by composition as above, the following required equations clearly hold:

$$A[1] = A \qquad A[\sigma][\tau] = A[\sigma \circ \tau] \qquad u[1] = u \qquad u[\sigma][\tau] = u[\sigma \circ \tau]$$

Context comprehension. Given a type $\Gamma \vdash A$, the C-space $\Gamma.A$ is defined by

$$\Gamma.A \;:\equiv\; \Sigma_{\gamma:\Gamma} A_\gamma$$

and $\mathrm{Probe}(\Gamma.A)$ by the condition that a map $p\colon \mathbf{2}^{\mathbb{N}} \to \Gamma.A$ is in $\mathrm{Probe}(\Gamma.A)$ iff $\mathrm{pr}_1 \circ p \in \mathrm{Probe}(\Gamma)$ and $\mathrm{pr}_2 \circ p \in \mathrm{Probe}(A)_{(\mathrm{pr}_1\circ p)}$, where pr_1 and pr_2 are projections of dependent sums. We skip the routine proof of the probe axioms.

The continuous map $\mathrm{p}\colon \Gamma.A \to \Gamma$ is the first projection pr_1, which is clearly continuous.

The term $\Gamma.A \vdash \mathrm{q} : A[\mathrm{p}]$ is the second projection pr_2, which clearly satisfies the condition of terms.

Given a continuous map $\sigma\colon \Delta \to \Gamma$ and a term $\Gamma \vdash u : A[\sigma]$, the extended substitution $(\sigma, u)\colon \Delta \to \Gamma.A$ is defined by

$$(\sigma, u)(\delta) \;:\equiv\; (\sigma(\delta), u(\delta))$$

for $\delta \in \Delta$, which is clearly continuous.

Using the universal property of dependent sums, it is routine to check the following equations:

$$\mathrm{p} \circ (\sigma, u) = \sigma \qquad \mathrm{q}[(\sigma, u)] = u \qquad (\mathrm{p}, \mathrm{q}) = 1 \qquad (\sigma, u) \circ \tau = (\sigma \circ \tau, u[\tau])$$

Σ-types. Given types $\Gamma \vdash A$ and $\Gamma.A \vdash B$, the type $\Gamma \vdash \Sigma AB$ is defined by

$$(\Sigma AB)_\gamma \;:\equiv\; \Sigma_{a:A_\gamma} B_{(\gamma,a)}$$

and $\mathrm{Probe}(\Sigma AB)$ by the condition that, for all $p \in P$, a map $q\colon \Pi_{\alpha:\mathbf{2}^{\mathbb{N}}} (\Sigma AB)_{p(\alpha)}$ is in $\mathrm{Probe}(\Sigma AB)_p$ iff $\mathrm{pr}_1 \circ q \in \mathrm{Probe}(A)_p$ and $\mathrm{pr}_2 \circ q \in \mathrm{Probe}(B)_{(p,\mathrm{pr}_1\circ q)}$.

Given terms $\Gamma \vdash u : A$ and $\Gamma \vdash v : B[(1, u)]$, the term $\Gamma \vdash (u, v) : \Sigma AB$ is defined by

$$(u, v) \colon \Pi_{\gamma:\Gamma}(\Sigma AB)_\gamma \qquad (u, v)(\gamma) :\equiv (u(\gamma), v(\gamma)).$$

Clearly (u, v) satisfies the condition of terms.

Given a term $\Gamma \vdash w : \Sigma AB$, the term $\Gamma \vdash \mathrm{pr}_1(w) : A$ is defined by

$$\mathrm{pr}_1(w) \colon \Pi_{\gamma:\Gamma} A_\gamma \qquad \mathrm{pr}_1(w)(\gamma) :\equiv \mathrm{pr}_1(w(\gamma))$$

and the term $\Gamma \vdash \mathrm{pr}_2(w) : B[(1, \mathrm{pr}_1(w))]$ by

$$\mathrm{pr}_2(w) \colon \Pi_{\gamma:\Gamma} B_{(\gamma, \mathrm{pr}_1(w(\gamma)))} \qquad \mathrm{pr}_2(w)(\gamma) :\equiv \mathrm{pr}_2(w(\gamma)).$$

Clearly both satisfy the condition of terms.

Using the universal property of dependent sums, one can easily verify the following equations:

$$\mathrm{pr}_1(u, v) = u \qquad \mathrm{pr}_2(u, v) = v \qquad (\mathrm{pr}_1(w), \mathrm{pr}_2(w)) = w$$

$$(\Sigma AB)[\sigma] = \Sigma(A[\sigma])(B[(\sigma \circ \mathrm{p}, \mathrm{q})])$$

$$(u, v)[\sigma] = (u[\sigma], v[\sigma]) \qquad \mathrm{pr}_1(w)[\sigma] = \mathrm{pr}_1(w[\sigma]) \qquad \mathrm{pr}_2(w)[\sigma] = \mathrm{pr}_2(w[\sigma])$$

Π-types. Given types $\Gamma \vdash A$ and $\Gamma.A \vdash B$, the type $\Gamma \vdash \Pi AB$ is defined as follows: Given $\gamma \in \Gamma$, we define $(\Pi AB)_\gamma$ to be the set of dependent functions $\varphi \colon \Pi_{a:A_\gamma} B_{(\gamma,a)}$ such that $\varphi \circ q \in \mathrm{Probe}(B)_{\lambda\alpha.(\gamma, q(\alpha))}$ whenever $q \in \mathrm{Probe}(A)_{\lambda\alpha.\gamma}$. Given $p \in \mathrm{Probe}(\Gamma)$, we say a map $r \colon \Pi_{\alpha:\mathbf{2}^\mathbb{N}}(\Pi AB)_{p(\alpha)}$ is in $\mathrm{Probe}(\Pi AB)_p$ iff $\lambda\alpha.r(t\alpha)(q\alpha) \in \mathrm{Probe}(B)_{\lambda\alpha.(p(t\alpha), q\alpha)}$ for all $t \in \mathrm{C}$ and $q \in \mathrm{Probe}(A)_{p \circ t}$. Now we verify the three conditions of types:

- Given $\gamma \in \Gamma$, $\varphi \in (\Pi AB)_\gamma$, $t \in \mathrm{C}$ and $q \in \mathrm{Probe}(A)_{\lambda\alpha.\gamma}$, the composite $\varphi \circ q$ is in $\mathrm{Probe}(B)_{\lambda\alpha.(\gamma, q(\alpha))}$ since φ is in $(\Pi AB)_\gamma$; thus $\lambda\alpha.\varphi$ is in $\mathrm{Probe}(\Pi AB)_{\lambda\alpha.\gamma}$.

- Given $p \in \mathrm{Probe}(\Gamma)$, $r \in \mathrm{Probe}(\Pi AB)_p$, $t, t' \in \mathrm{C}$ and $q \in \mathrm{Probe}(A)_{p \circ t \circ t'}$, we have $\lambda\alpha.r(t(t'\alpha))(q\alpha) \in \mathrm{Probe}(B)_{\lambda\alpha.(p(t(t'\alpha)), q\alpha)}$ by applying the proof of $r \in \mathrm{Probe}(\Pi AB)_p$ to $t \circ t'$ and q.

- Given $p \in \mathrm{Probe}(\Gamma)$, $r_0 \in \mathrm{Probe}(\Pi AB)_{p \circ \mathrm{cons}_0}$ and $r_1 \in \mathrm{Probe}(\Pi AB)_{p \circ \mathrm{cons}_1}$, we show that the map $r \colon \mathbf{2}^\mathbb{N} \to (\Pi AB)_{p(\alpha)}$, defined by $r(i\alpha) = r_i(\alpha)$, is in $\mathrm{Probe}(\Pi AB)_p$ as follows: Given $t \in \mathrm{C}$ and $q \in \mathrm{Probe}(A)_{p \circ t}$, we let $n = \mathrm{mod}_t(1)$. By (†), for each $s \in \mathbf{2}^n$, there are $i \in \mathbf{2}$ and $t' \in \mathrm{C}$ such that $t \circ \mathrm{cons}_s = \mathrm{cons}_i \circ t'$. Since $p \circ t \circ \mathrm{cons}_s = p \circ \mathrm{cons}_i \circ t'$, the composite $q \circ \mathrm{cons}_s \in \mathrm{Probe}(A)_{p \circ \mathrm{cons}_i \circ t'}$. Then we have

  $$(\lambda\alpha.r(t\alpha)(q\alpha)) \circ \mathrm{cons}_s = \lambda\alpha.r_i(t'\alpha)((q \circ \mathrm{cons}_s)\alpha)$$

  which is in $\mathrm{Probe}(B)_{(\lambda\alpha.(p(t\alpha), q\alpha)) \circ \mathrm{cons}_s}$, because $r_i \in \mathrm{Probe}(\Pi AB)_{p \circ \mathrm{cons}_i}$. Then using (t3') of B, we know that $\lambda\alpha.r(t\alpha)(q\alpha) \in \mathrm{Probe}(B)_{\lambda\alpha.(p(t\alpha), q\alpha)}$ and thus $r \in \mathrm{Probe}(\Pi AB)_p$.

Thus $\Gamma \vdash \Pi AB$ is well-defined.

Given a term $\Gamma.A \vdash v : B$, we define $\Gamma \vdash \lambda v : \Pi AB$ by

$$\lambda v \colon \Pi_{\gamma:\Gamma}(\Pi AB)_\gamma \qquad \lambda v(\gamma) :\equiv \lambda a.v(\gamma, a).$$

For $a \in A_\gamma$ the result $v(\gamma, a)$ is in $B_{(\gamma,a)}$. And, for any $q \in \mathrm{Probe}(A)_{\lambda\alpha.\gamma}$, we have $(\lambda a.v(\gamma, a))\circ q = v\circ(\lambda\alpha.(\gamma, q\alpha)) \in \mathrm{Probe}(B)_{\lambda\alpha.(\gamma,q\alpha)}$ because v is a term. Thus $\lambda a.v(\gamma, a) \in (\Pi AB)_\gamma$ and λv is well-defined. It remains to show that λv is a term: given $p \in \mathrm{Probe}(\Gamma)$, $t \in \mathrm{C}$ and $q \in \mathrm{Probe}(A)_{p\circ t}$, we have $\lambda\alpha.(\lambda v)(t\alpha)(q\alpha) = v \circ (\lambda\alpha.(p(t\alpha), q\alpha)) \in \mathrm{Probe}(B)_{\lambda\alpha.(p(t\alpha),q\alpha)}$ using the fact that v is a term.

Given terms $\Gamma \vdash w : \Pi AB$ and $\Gamma \vdash u : A$, the term $\Gamma \vdash \mathrm{app}(w, u) : B[(1, u)]$ is defined by

$$\mathrm{app}(w, u)\colon \Pi_{\gamma:\Gamma} B_{(\gamma,u(\gamma))} \qquad \mathrm{app}(w, u)(\gamma) :\equiv w(\gamma)(u(\gamma)).$$

Now we show that the condition of terms is satisfied: Given $p \in \mathrm{Probe}(\Gamma)$, we have $u\circ p \in \mathrm{Probe}(A)_p$ and $w\circ p \in \mathrm{Probe}(\Pi AB)_p$ since they are terms. Expanding the definition of $\mathrm{Probe}(\Pi AB)_p$ and using the uniform continuity of the identity map $1\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$, we know $\mathrm{app}(w, u) \circ p = \lambda\alpha.w(p\alpha)(u(p\alpha))$ is in $\mathrm{Probe}(B)_{(\lambda\gamma.(\gamma,u\gamma))\circ p}$.

It remains to verify the following required equations:

- $\mathrm{app}(\lambda v, u) = v[(1, u)]$

  For any $\Gamma.A \vdash v : B$, $\Gamma \vdash u : A$ and $\gamma \in \Gamma$, we have

  $$\begin{array}{lll} & \mathrm{app}(\lambda v, u)(\gamma) & \\ = & (\lambda v(\gamma))(u(\gamma)) & \text{(by the definition of app)} \\ = & v(\gamma, u(\gamma)) & \text{(by the definition of } \lambda) \\ = & v[(1, u)](\gamma). & \end{array}$$

- $\lambda(\mathrm{app}(w[\mathrm{p}], \mathrm{q})) = w$

  For any $\Gamma \vdash w : \Pi AB$, $\gamma \in \Gamma$ and $a \in A_\gamma$, we have

  $$\begin{array}{lll} & (\lambda(\mathrm{app}(w[\mathrm{p}], \mathrm{q})))(\gamma)(a) & \\ = & \mathrm{app}(w[\mathrm{p}], \mathrm{q})(\gamma, a) & \text{(by the definition of } \lambda) \\ = & w[\mathrm{p}](\gamma, a)(\mathrm{q}(\gamma, a)) & \text{(by the definition of app)} \\ = & w(\gamma)(a). & \end{array}$$

- $(\Pi AB)[\sigma] = \Pi(A[\sigma])(B[(\sigma \circ \mathrm{p}, \mathrm{q})])$

  For any $\Gamma \vdash A$, $\Gamma.A \vdash B$, $\sigma\colon \Delta \to \Gamma$ and $\delta \in \Delta$, we have

  $$\begin{array}{ll} & \Pi_{a:(A[\sigma])_\delta}(B[(\sigma \circ \mathrm{p}, \mathrm{q})])_{(\delta,a)} \\ = & \Pi_{a:A_{\sigma(\delta)}} B_{(\sigma\circ\mathrm{p},\mathrm{q})(\delta,a)} \\ = & \Pi_{a:A_{\sigma(\delta)}} B_{(\sigma(\delta),a)}. \end{array}$$

  Thus the elements of $(\Pi AB)_{\sigma(\delta)}$ and $(\Pi(A[\sigma])(B[(\sigma\circ\mathrm{p}, \mathrm{q})]))_\delta$ have the same type. It is easy to show that a map $\Pi_{a:A_{\sigma(\delta)}} B_{(\sigma(\delta),a)}$ is in $(\Pi AB)_{\sigma(\delta)}$ iff it is in $(\Pi(A[\sigma])(B[(\sigma\circ \mathrm{p}, \mathrm{q})]))_\delta$.

- $(\lambda v)[\sigma] = \lambda(v[(\sigma \circ \mathrm{p}, \mathrm{q})])$

  For any $\Gamma.A \vdash v : B$, $\sigma\colon \Delta \to \Gamma$, $\delta \in \Delta$ and $a \in A_{\sigma(\delta)}$, we have

  $$\begin{array}{ll} & (\lambda v)[\sigma](\delta)(a) \\ = & v(\sigma(\delta), a) \\ = & v[(\sigma \circ \mathrm{p}, \mathrm{q})](\delta, a) \\ = & \lambda(v[(\sigma \circ \mathrm{p}, \mathrm{q})])(\delta)(a). \end{array}$$

- $\mathrm{app}(w, u)[\sigma] = \mathrm{app}(w[\sigma], u[\sigma])$

For any $\Gamma \vdash w : \Pi AB$, $\Gamma \vdash u : A$, $\sigma \colon \Delta \to \Gamma$ and $\delta \in \Delta$, we have

$$\begin{aligned} & \mathrm{app}(w,u)[\sigma](\delta) \\ = \;& w(\sigma(\delta))(u(\sigma(\delta))) \\ = \;& (w[\sigma])(\delta)((u[\sigma])(\delta)) \\ = \;& \mathrm{app}(w[\sigma], u[\sigma])(\delta). \end{aligned}$$

6.5 A sheaf model of dependent types

Coquand presents a presheaf model of dependent types in his note [21], which is a category with families supporting Σ-types, Π-types and universes, using set theory as the meta-language. However, he has not extended his construction to sheaves. In this section, we construct a CwF-structure on the category of sheaves on our uniform-continuity site (defined in Chapter 3.2.1), following Coquand's method [21].

In this section, we write Γ, Δ, Θ to denote sheaves, as they represent contexts, σ, τ, ν to denote natural transformations, as they represent substitutions, A, B to denote (interpretations of) types, and u, v, w to denote (interpretations of) terms.

The base category. Recall that a *sheaf* is a (small) set Γ equipped with an *action*

$$\_ \cdot \_ : \; \Gamma \to \mathrm{C} \to \Gamma$$

satisfying the following conditions:

(s1) $\gamma \cdot 1 = \gamma$ for all $\gamma \in \Gamma$,

(s2) $(\gamma \cdot t) \cdot r = \gamma \cdot (t \circ r)$ for all $\gamma \in \Gamma$ and $t, r \in \mathrm{C}$, and

(s3) for any γ_0, γ_1 there exists a unique amalgamation $\gamma \in \Gamma$ such that $\gamma \cdot \mathrm{cons}_0 = \gamma_0$ and $\gamma \cdot \mathrm{cons}_1 = \gamma_1$.

As discussed in Chapter 3.2.1, the condition (s3) is logically equivalent to the following:

(s3') for any $n \in \mathbb{N}$ and $\{\gamma_s\}_{s \in \mathbf{2}^n}$, there exists a unique amalgamation $\gamma \in \Gamma$ such that $\gamma \cdot \mathrm{cons}_s = \gamma_s$ for all $s \in \mathbf{2}^n$.

A map $\sigma \colon \Delta \to \Gamma$ of sheaves is a *natural transformation* if

$$\sigma(\delta) \cdot t = \sigma(\delta \cdot t)$$

for all $\delta \in \Delta$ and $t \in \mathrm{C}$.

We abbreviate Γ for the sheaf $(|\Gamma|, \cdot)$, where $|\Gamma|$ stands for the underlying set and $\cdot$ for the action, and often write Γ to mean $|\Gamma|$.

Sheaves and natural transformations defined as above form the base category of the model, i.e. contexts are interpreted as sheaves and substitutions as natural transformations. In particular, the terminal object (or empty context) is a singleton sheaf.

Types and terms. A *type* $\Gamma \vdash A$ is a Γ-indexed family of sets, i.e. $A = \{A_\gamma\}_{\gamma \in \Gamma}$ where each A_γ is a (small) set, equipped with a *restriction map*

$$\_ * \_ : \; A_\gamma \to \prod_{t \colon \mathrm{C}} A_{\gamma \cdot t}$$

for every $\gamma \colon \Gamma$, satisfying the following conditions:

(t1) $a * 1 = a$ for all $\gamma \in \Gamma$ and $a \in A_\gamma$,

(t2) $(a * t) * r = a * (t \circ r)$ for all $\gamma \in \Gamma$, $a \in A_\gamma$ and $t, r \in \mathrm{C}$, and

(t3) for any $\gamma \in \Gamma$, $a_0 \in A_{\gamma \cdot \mathrm{cons}_0}$ and $a_1 \in A_{\gamma \cdot \mathrm{cons}_1}$, there is a unique amalgamation $a \in A_\gamma$ such that $a * \mathrm{cons}_0 = a_0$ and $a * \mathrm{cons}_1 = a_1$.

Intuitively, a type over Γ is a dependent "sheaf" which depends on the sheaf Γ. Similarly to the logical equivalence of (s3) and (s3'), the condition (t3) is logically equivalent to

(t3') for any $\gamma \in \Gamma$, $n \in \mathbb{N}$ and $\{a_s \in A_{\gamma \cdot \mathrm{cons}_s}\}_{s \in \mathbf{2}^n}$, there is a unique amalgamation $a \in A_\gamma$ such that $a * \mathrm{cons}_s = a_s$ for all $s \in \mathbf{2}^n$.

Given a type $\Gamma \vdash A$, a *term* $\Gamma \vdash u : A$ is a dependent function

$$u \colon \prod_{\gamma : \Gamma} A_\gamma$$

such that

$$u(\gamma) * t = u(\gamma \cdot t)$$

for all $\gamma \in \Gamma$ and $t \in \mathrm{C}$. The intuition of the above condition is that terms are "dependent natural transformations".

The following lemma of equality over elements of a type is analogous to Lemma 3.2.5:

Lemma 6.5.1. *Let A be a type over Γ. Given $\gamma \in \Gamma$ and $a, a' \in A_\gamma$, the equation $a = a'$ holds iff there exists $n \in \mathbb{N}$ such that $a * \mathrm{cons}_s = a' * \mathrm{cons}_s$ for all $s \in \mathbf{2}^n$.*

Proof. ($\Rightarrow$) is obvious. ($\Leftarrow$) uses condition (t3'). □

Substitutions of types and terms. Given a type $\Gamma \vdash A$ and a substitution $\sigma \colon \Delta \to \Gamma$, the *substituted type* $\Delta \vdash A[\sigma]$ is defined by composition (or reindexing), i.e. $(A[\sigma])_\delta :\equiv A_{\sigma(\delta)}$ for all $\delta \in \Delta$, and restriction maps of $A[\sigma]$ are inherited from A. We can easily verify that $A[\sigma]$ satisfies conditions (t1) to (t3), using the naturality of σ. Since type substitutions are defined by composition, the following equations hold:

$$A[1] = A, \qquad A[\sigma][\tau] = A[\sigma \circ \tau].$$

Given a term $\Gamma \vdash u : A$ and a substitution $\sigma \colon \Delta \to \Gamma$, the substituted term $\Delta \vdash u[\sigma] : A[\sigma]$ is again defined by composition, i.e. $u[\sigma] :\equiv u \circ \sigma$. The above condition is satisfied because of the naturality of σ. Since term substitutions are defined by composition, the following equations hold:

$$u[1] = u, \qquad u[\sigma][\tau] = u[\sigma \circ \tau].$$

Context comprehension. Given a type $\Gamma \vdash A$, we construct a new sheaf $\Gamma.A$ by the dependent sum

$$\Gamma.A \;:\equiv\; \sum_{\gamma \colon \Gamma} A_\gamma.$$

The action is defined by, for all $(\gamma, a) \in \Gamma.A$ and $t \in \mathrm{C}$,

$$(\gamma, a) \cdot t \;:\equiv\; (\gamma \cdot t, a * t).$$

Then we have to verify that the three conditions (s1) to (s3) of sheaves are satisfied, which is routine.

The natural transformation $\mathrm{p}\colon \Gamma.A \to \Gamma$ is given by the first projection

$$\mathrm{p}\colon \sum_{\gamma:\Gamma} A_\gamma \to \Gamma \qquad \mathrm{p}(\gamma, u) :\equiv \gamma.$$

And the term $\Gamma.A \vdash \mathrm{q} : A[\mathrm{p}]$ is given by the second projection

$$\mathrm{q}\colon \prod_{w:\sum_{\gamma:\Gamma} A_\gamma} A_{\mathrm{p}(w)} \qquad \mathrm{q}(\gamma, u) :\equiv u.$$

Clearly, the map p is a natural transformation, and q satisfies the condition of term.

Substitution extension is defined using the universal property of dependent sums: Given a natural transformation $\sigma\colon \Delta \to \Gamma$ and a term $\Delta \vdash b : A[\sigma]$, we define the map

$$(\sigma, b)\colon \Delta \to \sum_{\gamma:\Gamma} A_\gamma \qquad \text{by} \qquad (\sigma, b)(\delta) = (\sigma(\delta), b(\delta))$$

which is clearly a natural transformation $\Delta \to \Gamma.A$. It satisfies the required universal property, i.e. (σ, b) is the unique (up to pointwise equality) natural transformation such that the equations $\mathrm{p} \circ (\sigma, b) = \sigma$ and $\mathrm{q}[(\sigma, b)] = b$ hold (pointwise), since p and q are given by the projection maps.

Σ-types. Given types $\Gamma \vdash A$ and $\Gamma.A \vdash B$, we define $\Gamma \vdash \Sigma AB$ by, for each $\gamma \in \Gamma$

$$(\Sigma AB)_\gamma \;:\equiv\; \sum_{a:A_\gamma} B_{(\gamma,a)}.$$

The restriction map on $\gamma \in \Gamma$ is defined componentwise, i.e. for any $(a, b) \in (\Sigma AB)_\gamma$ and $t \in \mathrm{C}$,

$$(a, b) * t \;:\equiv\; (a * t, b * t).$$

It is routine to show that the conditions (t1) to (t3) are satisfied.

Given terms $\Gamma \vdash u : A$ and $\Gamma \vdash v : B[(1, u)]$, we define the term $\Gamma \vdash (u, v) : \Sigma AB$ by

$$(u, v)\colon \prod_{\gamma:\Gamma} (\Sigma AB)_\gamma \qquad (u, v)(\gamma) :\equiv (u(\gamma), v(\gamma)).$$

Since $v(\gamma) \in B_{(\gamma,u(\gamma))}$, the pair $(u, v)(\gamma)$ is well defined. One can easily show that (u, v) satisfies the condition of terms by following the definitions.

Projections are defined via the projection maps in the meta-theory: Given a term $\Gamma \vdash w : \Sigma AB$, we define the term $\Gamma \vdash \mathrm{pr}_1(w) : A$ by

$$\mathrm{pr}_1(w)\colon \prod_{\gamma:\Gamma} A_\gamma \qquad (\mathrm{pr}_1(w))(\gamma) :\equiv \mathrm{pr}_1(w(\gamma)),$$

and the term $\Gamma \vdash \mathrm{pr}_2(w) : B[(1, \mathrm{pr}_1(w))]$ by

$$\mathrm{pr}_2(w) \colon \prod_{\gamma:\Gamma} B_{(\gamma, \mathrm{pr}_1(w(\gamma)))} \qquad (\mathrm{pr}_2(w))(\gamma) :\equiv \mathrm{pr}_2(w(\gamma)),$$

where the rightmost pr_1 and pr_2 are the projections in the meta-theory. It is routine to show that $\mathrm{pr}_1(w)$ and $\mathrm{pr}_2(w)$ satisfy the condition of terms.

Moreover, we need to verify the following required equations, which is routine:

1. $(\mathrm{pr}_1(w), \mathrm{pr}_2(w)) = w$,
2. $\mathrm{pr}_1(u, v) = u$,
3. $\mathrm{pr}_2(u, v) = v$,
4. $(\Sigma AB)[\sigma] = \Sigma(A[\sigma])(B[(\sigma \circ \mathrm{p}, \mathrm{q})])$,
5. $(u, v)[\sigma] = (u[\sigma], v[\sigma])$,
6. $\mathrm{pr}_1(w)[\sigma] = \mathrm{pr}_1(w[\sigma])$,
7. $\mathrm{pr}_2(w)[\sigma] = \mathrm{pr}_2(w[\sigma])$.

Notice that the equations of terms hold pointwise while the one of types holds up to isomorphism.

Π-types. Given types $\Gamma \vdash A$ and $\Gamma.A \vdash B$, we define $\Gamma \vdash \Pi AB$ by, for each $\gamma \in \Gamma$, choosing $(\Pi AB)_\gamma$ to be the set of dependent functions $\varphi \colon \prod_{t:\mathrm{C}} \prod_{a:A_{\gamma \cdot t}} B_{(\gamma \cdot t, a)}$ such that

$$\varphi_t(a) * r = \varphi_{t \circ r}(a * r)$$

for any $t, r \in \mathrm{C}$ and $a \in A_{\gamma \cdot t}$. One can think of the elements of $(\Pi AB)_\gamma$ as C-indexed families of terms of type B. (This condition is necessary because app should satisfy the condition of terms as shown later.) The restriction map on $\gamma \in \Gamma$ is defined by, for any $\varphi \in (\Pi AB)_\gamma$ and $t, r \in \mathrm{C}$,

$$(\varphi * t)_r \ :\equiv \ \varphi_{(t \circ r)}.$$

Following the definitions, we know that $\varphi * t$ is in $(\Pi AB)_{\gamma \cdot t}$ if φ is in $(\Pi AB)_\gamma$; thus the restriction map is well-defined. Conditions (t1) and (t2) are easy to verify. Here we show that (t3) is also satisfied: Given $\gamma \in \Gamma$, $\varphi^0 \in (\Pi AB)_{\gamma \cdot \mathrm{cons}_0}$ and $\varphi^1 \in (\Pi AB)_{\gamma \cdot \mathrm{cons}_1}$, we define a dependent map $\varphi \colon \prod_{t:\mathrm{C}} \prod_{a:A_{\gamma \cdot t}} B_{(\gamma \cdot t, a)}$ as follows: given $t \in \mathrm{C}$ and $a \in A_{\gamma \cdot t}$, we let n to be the least $\mathrm{mod}_t(1)$ (see Chapter 3.2.1 for the definition of mod_t). For each $s \in \mathbf{2}^n$ we have $a * \mathrm{cons}_s \in A_{\gamma \cdot \mathrm{cons}_s}$. Since $t \cdot \mathrm{cons}_s = \mathrm{cons}_{i_s} \cdot t_s$ for some $i_s \in \mathbf{2}$ and $t_s \in \mathrm{C}$, the element $a * \mathrm{cons}_s$ is also in $A_{\gamma \cdot \mathrm{cons}_{i_s} \cdot t_s}$ and thus we have $\varphi^{i_s}_{t_s}(a * \mathrm{cons}_s) \in B_{(\gamma \cdot \mathrm{cons}_{i_s} \cdot t_s, a * \mathrm{cons}_s)}$. Because $(\gamma \cdot \mathrm{cons}_{i_s} \cdot t_s, a * \mathrm{cons}_s) = (\gamma \cdot t \cdot \mathrm{cons}_s, a * \mathrm{cons}_s) = (\gamma \cdot t, a) * \mathrm{cons}_s$, we get a family $\{\varphi^{i_s}_{t_s}(a * \mathrm{cons}_s) \in B_{(\gamma \cdot t, a) * \mathrm{cons}_s}\}_{s \in \mathbf{2}^n}$. Using (t3') of type B, we define $\varphi_t(a)$ to be the unique amalgamation of that family, and have

$$\forall s \in \mathbf{2}^n.\ \varphi_t(a) * \mathrm{cons}_s = \varphi^{i_s}_{t_s}(a * \mathrm{cons}_s) \qquad (\ddagger)$$

Now we have to prove:

(1) φ is in $(\Pi AB)_\gamma$: Given $t, r \in \mathrm{C}$ and $a \in A_{\gamma\cdot t}$, we let n be the least $\mathrm{mod}_t(1)$ and m be the least $\mathrm{mod}_r(n)$ and have

$$\forall s \in \mathbf{2}^m.\ \exists s' \in \mathbf{2}^n.\ \exists r' \in \mathrm{C}.\ \exists i \in \mathbf{2}.\ \exists t' \in \mathrm{C}.\ t \circ r \circ \mathrm{cons}_s = t \circ \mathrm{cons}'_s \circ r' = \mathrm{cons}_i \circ t' \circ r'$$

by applying (†) twice. For each $s \in \mathbf{2}^m$, we have

$$\begin{array}{rcll}
\varphi_t(a) * r * \mathrm{cons}_s & = & \varphi_t(a) * \mathrm{cons}_{s'} * r' & (\text{by } (\dagger)) \\
& = & \varphi^i_{t'}(a * \mathrm{cons}_{s'}) * r' & (\text{by } (\ddagger)) \\
& = & \varphi^i_{(t' \circ r')}(a * \mathrm{cons}_{s'} * r') & (\varphi^i \text{ is in } (\Pi AB)_\gamma) \\
& = & \varphi_{(\mathrm{cons}_i \circ t' \circ r')}(a * \mathrm{cons}_{s'} * r') & (\varphi \text{ is an amalgamation (see (2))}) \\
& = & \varphi_{(t \circ r \circ \mathrm{cons}_s)}(a * r * \mathrm{cons}_s). & (\text{by } (\dagger))
\end{array}$$

Let k be the least $\mathrm{mod}_{(t\circ r)}(1)$. We know $k \leq m$ and thus let $s = s_0 s_1$ where $s_0 \in \mathbf{2}^k$ and $s_1 \in \mathbf{2}^{(m-k)}$. Then we have $t \circ r \circ \mathrm{cons}_{s_0} = \mathrm{cons}_i \circ u$ for some $u \in \mathrm{C}$. (Notice that i is the same as in the above equation. We omit the proof as this fact is not needed to get the following equation.) Now we also have

$$\begin{array}{rcll}
\varphi_{(t\circ r)}(a * r) * \mathrm{cons}_s & = & \varphi_{(t\circ r)}(a * r) * \mathrm{cons}_{s_0} * \mathrm{cons}_{s_1} & (s = s_0 s_1) \\
& = & \varphi^i_u(a * r * \mathrm{cons}_{s_0}) * \mathrm{cons}_{s_1} & (\text{by } (\ddagger)) \\
& = & \varphi^i_{(u \circ \mathrm{cons}_{s_1})}(a * r * \mathrm{cons}_{s_0} * \mathrm{cons}_{s_1}) & (\varphi^i \text{ is in } (\Pi AB)_\gamma) \\
& = & \varphi_{(\mathrm{cons}_i \circ u \circ \mathrm{cons}_{s_1})}(a * r * \mathrm{cons}_s) & (\varphi \text{ is an amalgamation}) \\
& = & \varphi_{(t \circ r \circ \mathrm{cons}_s)}(a * r * \mathrm{cons}_s). & (\text{by } (\dagger))
\end{array}$$

Therefore, we have $\varphi_t(a) * r * \mathrm{cons}_s = \varphi_{(t\circ r)}(a * r) * \mathrm{cons}_s$ for all $s \in \mathbf{2}^m$. Applying Lemma 6.5.1 gives the desired result.

(2) φ is an amalgamation of φ^0 and φ^1: Given $t \in \mathrm{C}$, $i \in \{0, 1\}$ and $a \in A_{\gamma\cdot\mathrm{cons}_i\cdot t}$, we know 0 is the least $\mathrm{mod}_{\mathrm{cons}_i\cdot t}(1)$, and thus have

$$\begin{array}{rcll}
(\varphi * \mathrm{cons}_i)_t(a) & = & \varphi_{(\mathrm{cons}_i \circ t)}(a) & (\text{by the definition of } _ * _) \\
& = & \varphi_{(\mathrm{cons}_i \circ t)}(a) * \mathrm{cons}_\varepsilon & (\mathrm{cons}_\varepsilon \equiv 1) \\
& = & \varphi^i_t(a). & (\text{by } (\ddagger))
\end{array}$$

(3) φ is unique (up to pointwise equality): Suppose $\psi \in (\Pi AB)_\gamma$ is an amalgamation of φ^0 and φ^1. Given $t \in \mathrm{C}$ and $a \in A_{\gamma\cdot t}$, we let n be the least $\mathrm{mod}_t(1)$ and have, for each $s \in \mathbf{2}^n$,

$$\begin{array}{rcll}
\psi_t(a) * \mathrm{cons}_s & = & \psi_{(t \circ \mathrm{cons}_s)}(a * \mathrm{cons}_s) & (\psi \text{ is in } (\Pi AB)_\gamma) \\
& = & \psi_{(\mathrm{cons}_{i_s} \circ t_s)}(a * \mathrm{cons}_s) & (\text{by } (\dagger), \text{ i.e. } t \circ \mathrm{cons}_s = \mathrm{cons}_{i_s} \circ t_s) \\
& = & (\psi * \mathrm{cons}_{i_s})_{t_s}(a * \mathrm{cons}_s) & (\text{by the definition of } _ * _) \\
& = & \varphi^{i_s}_{t_s}(a * \mathrm{cons}_s) & (\psi \text{ is an amalgamation}) \\
& = & \varphi_t(a) * \mathrm{cons}_s & (\text{by } (\ddagger))
\end{array}$$

and thus $\psi_t(a) = \varphi_t(a)$ by Lemma 6.5.1.

Given a term $\Gamma.A \vdash v : B$, the abstraction $\Gamma \vdash \lambda v : \Pi AB$ is defined by

$$\lambda v\colon \prod_{\gamma:\Gamma} (\Pi AB)_\gamma \qquad ((\lambda v)(\gamma))_t(a) :\equiv v(\gamma \cdot t, a).$$

We know $v\colon \prod_{w:\sum_{\gamma:\Gamma} A_\gamma} B_w$. Then $v(\gamma \cdot t, a) \in B_{(\gamma \cdot t, a)}$, and thus $(\lambda v)(\gamma)$ is well-defined. Now we show that $(\lambda v)(\gamma)$ is in $(\Pi AB)_\gamma$: given $t, r \in \mathrm{C}$ and $a \in A_{\gamma \cdot t}$, we have

$$((\lambda v)(\gamma))_t(a) * r = (v(\gamma \cdot t, a)) * r = v(\gamma \cdot t \cdot r, a * r) = ((\lambda b)(\gamma))_{(t \circ r)}(a * r).$$

It remains to show that λv satisfies the condition of terms: given $\gamma \in \Gamma$, $t, r \in \mathrm{C}$ and $a \in A_{\gamma \cdot t \cdot r}$, we have

$$((\lambda v)(\gamma) * t)_r(a) = ((\lambda v)(\gamma))_{(t \circ r)}(a) = v(\gamma \cdot t \cdot r, a) = ((\lambda v)(\gamma \cdot t))_r(a)$$

and thus $(\lambda v)(\gamma) * t = (\lambda v)(\gamma \cdot t)$ holds pointwise.

Given terms $\Gamma \vdash w : \Pi AB$ and $\Gamma \vdash u : A$, the application $\Gamma \vdash \mathrm{app}(w, u) : B[(1, u)]$ is defined by

$$\mathrm{app}(w, u)\colon \prod_{\gamma:\Gamma} B_{(\gamma, u(\gamma))} \qquad (\mathrm{app}(w, u))(\gamma) :\equiv (w(\gamma))_1(u(\gamma)).$$

Since $(w(\gamma))_1\colon \prod_{a:A_\gamma} B_{(\gamma,a)}$ and $u(\gamma) \in A_\gamma$, the result $(\mathrm{app}(w, u))(\gamma)$ is well-defined. Now we show that $\mathrm{app}(w, u)$ satisfies the condition of terms: given $\gamma \in \Gamma$ and $t \in \mathrm{C}$, we have

$$\begin{array}{rcll}
(\mathrm{app}(w, u))(\gamma) * t & = & (w(\gamma))_1(u(\gamma)) * t & \text{(by the definition of app)} \\
& = & (w(\gamma))_t(u(\gamma) * t) & (w(\gamma) \text{ is in } (\Pi AB)_\gamma) \\
& = & (w(\gamma) * t)_1(u(\gamma) * t) & \text{(by the definition of } * \text{ on } \Pi AB) \\
& = & (w(\gamma \cdot t))_1(u(\gamma \cdot t)) & \text{(both } w \text{ and } u \text{ are terms)} \\
& = & (\mathrm{app}(w, u))(\gamma \cdot t). & \text{(by the definition of app)}
\end{array}$$

As mentioned earlier, the condition of elements in $(\Pi AB)_\gamma$ is used in the above proof that $\mathrm{app}(w, u)$ is a term.

Moreover, we need to verify the following equations. The equations of terms hold pointwise, while the one of types holds up to isomorphism.

1. $\mathrm{app}(\lambda v, u) = v[(1, u)]$

   $$\begin{array}{rcll}
   & \mathrm{app}(\lambda v, u)(\gamma) & & \\
   = & ((\lambda v)(\gamma))_1(u(\gamma)) & \text{(by the definition of app)} \\
   = & v(\gamma \cdot 1, u(\gamma)) & \text{(by the definition of } \lambda) \\
   = & v[(1, u)](\gamma) & \text{(by the definition of substitution extension)}
   \end{array}$$

   for any $\Gamma \vdash u : A$, $\Gamma.A \vdash v : B$ and $\gamma \in \Gamma$.

2. $\lambda(\mathrm{app}(w[\mathrm{p}], \mathrm{q})) = w$

$$
\begin{array}{lll}
 & (\lambda(\mathrm{app}(w[\mathrm{p}], \mathrm{q}))(\gamma))_t(a) & \\
= & \mathrm{app}(w[\mathrm{p}], \mathrm{q})(\gamma \cdot t, a) & \text{(by the definition of } \lambda) \\
= & (w[\mathrm{p}](\gamma \cdot t, a))_1(\mathrm{q}(\gamma \cdot t, a)) & \text{(by the definition of app)} \\
= & (w(\gamma \cdot t))_1(a) & \text{(by the definitions of p and q)} \\
= & (w(\gamma) * t)_1(a) & \text{(by the condition of term)} \\
= & (w(\gamma))_t(a) & \text{(by the definition of } * \text{ on } \Pi AB)
\end{array}
$$

for any $\Gamma \vdash w : \Pi AB$, $\gamma \in \Gamma$, $t \in \mathrm{C}$ and $a \in A_{\gamma \cdot t}$.

3. $(\Pi AB)[\sigma] = \Pi(A[\sigma])(B[(\sigma \circ \mathrm{p}, \mathrm{q})])$

   Given $\delta \in \Delta$, the families $(\Pi AB)_{\sigma(\delta)}$ and $(\Pi(A[\sigma])(B[(\sigma \circ \mathrm{p}, \mathrm{q})]))_\delta$ are isomorphic: elements of the first family have type $\Pi_{t:\mathrm{C}}\Pi_{a:A_{\sigma(\delta)\cdot t}} B_{(\sigma(\delta)\cdot t, a)}$, and ones of the second have type $\Pi_{t:\mathrm{C}}\Pi_{a:A_{\sigma(\delta\cdot t)}} B_{(\sigma(\delta\cdot t), a)}$. Due to the naturality of σ, i.e. $\sigma(\delta) \cdot t = \sigma(\delta \cdot t)$, elements of the two families have the same types, and the conditions that they satisfy are also equivalent.

4. $(\lambda v)[\sigma] = \lambda(v[(\sigma \circ \mathrm{p}, \mathrm{q}])$

$$
\begin{array}{lll}
 & ((\lambda v)[\sigma](\delta))_t(a) & \\
= & ((\lambda v)(\sigma(\delta)))_t(a) & \\
= & v(\sigma(\delta) \cdot t, a) & \text{(by the definition of } \lambda) \\
= & v(\sigma(\delta \cdot t), a) & \text{(by the naturality of } \sigma) \\
= & v[(\sigma \circ \mathrm{p}, \mathrm{q})](\delta \cdot t, a) & \\
= & (\lambda(v[(\sigma \circ \mathrm{p}, \mathrm{q}])))(\delta))_t(a) &
\end{array}
$$

for any $\Gamma.A \vdash v : B$, $\sigma \colon \Delta \to \Gamma$, $\delta \in \Delta$, $t \in \mathrm{C}$ and $a \in A_{\sigma(\delta)\cdot t}$.

5. $\mathrm{app}(w, u)[\sigma] = \mathrm{app}(w[\sigma], u[\sigma])$

$$
\begin{array}{lll}
 & \mathrm{app}(w, u)[\sigma](\delta) & \\
= & \mathrm{app}(w, u)(\sigma(\delta)) & \\
= & (w(\sigma(\delta)))_1(u(\sigma(\delta))) & \text{(by the definition of app)} \\
= & (w[\sigma](\delta))_1(u[\sigma](\delta)) & \\
= & \mathrm{app}(w[\sigma], u[\sigma])(\delta) &
\end{array}
$$

for any $\Gamma \vdash w : \Pi AB$, $\Gamma \vdash u : A$, $\sigma \colon \Delta \to \Gamma$ and $\delta \in \Delta$.

CHAPTER 7

Construction of the model in type theory

In the previous chapters we constructed a model *of* type theory in informal set theory. In this chapter, we discuss the construction of the model *in* Martin-Löf type theory, which we have formalized in Agda notation [11, 12, 63]. The main purpose of this formalization is to extract computational content from our model of C-spaces, as illustrated in Section 8.3, rather than merely certify that our constructions and proofs are correct.

The main difficulties of formulating the constructions and of proving theorems in type theory involve the presence of proof relevance and the lack of function extensionality in MLTT. For instance, a continuous map is formulated as a pair consisting of a underlying map and a continuity witness. In the informal proof of the fact that the domain $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$ of the fan functional is a discrete C-space (Lemma 3.5.1), we ignored continuity witnesses and proved equality of underlying maps only (using function extensionality implicitly). But in MLTT both equalities of underlying maps and of continuity witnesses are required. Even worse, a map $f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ can have more than one uniform-continuity witnesses, because if $m \in \mathbb{N}$ is a modulus of uniform continuity of f then so is any number that is greater than m. In order to formalize the proof of Lemma 3.5.1, we require the existence of a minimal modulus of uniform continuity. With this refinement, we still need function extensionality to prove that uniform continuity is a *proposition*, that is a type with at most one element.

The following results of the thesis have been formalized [76]:

1. The two formulations

   $$\Pi(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \|\Sigma(m{:}\mathbb{N}).\ \Pi(\alpha,\beta{:}\mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta\|$$
   $$\Pi(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \Sigma(m{:}\mathbb{N}).\ \Pi(\alpha,\beta{:}\mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta$$

   of the uniform-continuity principle are logically equivalent in intensional MLTT extended with propositional truncation.

   This theorem is already formulated and proved in type theory in Chapter 2.3, and the Agda proof is a direct translation.

2. C-Spaces form a (locally) cartesian closed category with a natural numbers object, a coproduct $\mathbf{1}+\mathbf{1}$, and a continuous functional fan$\colon (\mathbf{2}^{\mathbb{N}} \to \mathbb{N}) \to \mathbb{N}$ that calculates minimal moduli of uniform continuity (Chapters 3.3–3.5).

3. Assuming the Brouwerian principle that all type-theoretic functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous, the full type hierarchy is equivalent to the Kleene–Kreisel continuous hierarchy within C-spaces (Chapter 4.3).

4. C-Spaces give a model of Gödel's system T. All T-definable functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous. The uniform-continuity principle, expressed as a skolemization, is validated by the fan functional (Chapter 5.1).

5. C-Spaces give a continuous realizability semantic of HA^{ω}. The uniform-continuity principle is realized by the fan functional (Chapter 5.2).

6. The uniform-continuity principle in MLTT, expressed as a type via the Curry–Howard interpretation, is validated by the fan functional in the locally cartesian closed category of C-spaces, following Seely's method (Chapter 6.2).

In addition, we implemented several models of dependent types via the notion of category with families (CwF) discussed in Chapters 6.3–6.5. The purposes of formalizing the CwF of types/sets are to understand the CwF-structure of identity types, and to verify the equations with substitutions that we proposed for identity types in Chapter 6.3. We also formalized the basic CwF-structures of C-spaces and of (pre)sheaves, in order to explore the feasibility of type-theoretic implementation of sheaf models.

To discuss the type-theoretic development of the above work, we present some necessary constructions and theorems using informal type theory as in the HoTT book [72]. The basic notations have been introduced in Chapter 6.1. In particular, we also need the type of (small) types, that is the *universe* $\mathcal{U}$, *e.g.* to formulate C-spaces (see Section 7.2.1).

This chapter is organized as follows. Section 7.1 discusses the issues in type-theoretic development caused by the absence of function extensionality and by the presence of proof relevance in MLTT. Then Section 7.2 explains a few approaches, together with the main constructions and adjustments, to address those issues. The last part of this chapter, Section 7.3, demonstrates some experiments of internalizing models of dependent types.

7.1 Function extensionality and proof relevance

As mentioned above, the first difficulty of the type-theoretic development of the model is the absence of *function extensionality* (funext), which can be formulated as the following type

$$\Pi(X{:}\mathcal{U}).\ \Pi(Y{:}X{\to}\mathcal{U}).\ \Pi(f,g{:}\Pi(x{:}X).Y(x)).\ (\Pi(x{:}X).fx = gx) \to f = g,$$

in intensional MLTT. Another difficulty is caused by the fact that MLTT is *proof-relevant*, *i.e.* a proof in MLTT is a term of some type. Some issues related to such difficulties, discussed below, arise in the type-theoretic rendering of

(1) discrete C-spaces,

(2) exponentials of C-spaces, and

(3) the fan functional.

(1) and (2) are in a similar situation related to the lack of (funext) for functions $\mathbb{N} \to \mathbf{2}$. (3) is subtler and more interesting: (funext) is seemly necessary, but also not sufficient for constructing the fan functional and for proving its desired property, due to the presence of proof relevance in MLTT.

To discuss the above issues, we need the following types and functions:

- The cantor space
$$\mathbf{2}^{\mathbb{N}} : \mathcal{U}$$
is represented as the function type $\mathbb{N} \to \mathbf{2}$, where $\mathbb{N}$ denotes the type of natural numbers and $\mathbf{2}$ denotes the type of booleans as usual. Given $\alpha\colon \mathbf{2}^{\mathbb{N}}$ and $i\colon \mathbb{N}$, we write α_i to denote the application $\alpha(i)$.

- The less-than-or-equal-to relation of natural numbers
$$n \leq m : \mathcal{U}$$
is defined inductively with (i) a constructor $0 \leq m$ for each $m\colon \mathbb{N}$, and (ii) a function $n \leq m \to n+1 \leq m+1$. An equivalent definition of $n \leq m$ is $\Sigma(k{:}\mathbb{N}).\ n + k = m$, but the inductive definition is more convenient for our purposes.

- Given $\alpha, \beta\colon \mathbf{2}^{\mathbb{N}}$, the relation
$$\alpha =_n \beta : \mathcal{U}$$
is defined by induction on the natural number n, with (i) a constructor $\alpha =_0 \beta$, and (ii) a function $\alpha =_n \beta \to \alpha_n = \beta_n \to \alpha =_{n+1} \beta$. An equivalent definition of $\alpha =_n \beta$ is $\Pi(i{:}\mathbb{N}).\ i < n \to \alpha_i = \beta_i$, but again the inductive definition is more convenient for our purposes.

- The collection of binary sequences of length n
$$\mathbf{2}^n : \mathcal{U}$$
is represented as the vector type $\mathrm{Vec}(\mathbf{2}, n)$ whose inductive definition is available in Section 8.1 in Agda notation. We write $\varepsilon\colon \mathrm{Vec}(X, 0)$ to denote the empty vector, and $x :: xs\colon \mathrm{Vec}(X, n+1)$ to denote the concatenation of $x\colon X$ and $xs\colon \mathrm{Vec}(X, n)$.

- A property of elements of a type X is expressed as a type family $P\colon X \to \mathcal{U}$. We write $x \in P$ to denote $P(x)$. Following this idea, a naive formulation of uniform continuity of maps $\mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ is the the type family $\mathrm{C}\colon (\mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}) \to \mathcal{U}$ defined by
$$\mathrm{C}(t) :\equiv \Pi(m{:}\mathbb{N}).\ \Sigma(n{:}\mathbb{N}).\ \Pi(\alpha, \beta : \mathbf{2}^{\mathbb{N}}).\ \alpha =_n \beta \to t\alpha =_m t\beta.$$
Then a witness of uniform continuity of t is an element of type $t \in \mathrm{C}$.

- The concatenation map of binary sequences
$$\mathrm{cons}\colon \Pi(n{:}\mathbb{N}).\ \mathbf{2}^n \to \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$$
is defined by induction on finite sequences: (i) $\mathrm{cons}_\varepsilon \alpha :\equiv \alpha$, (ii) $(\mathrm{cons}_{(b::s)}\alpha)_0 :\equiv b$, and (iii) $(\mathrm{cons}_{(b::s)}\alpha)_{i+1} :\equiv (\mathrm{cons}_s \alpha)_i$. We omit the length n and simply write $\mathrm{cons}_s\alpha$ (rather than $\mathrm{cons}(n, s, \alpha)$) for the sake of readability.

- We also need the following functions

$$\mathrm{take}\colon \Pi(n{:}\mathbb{N}).\ \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^n \qquad \mathrm{drop}\colon \mathbb{N} \to \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}.$$

  The first function returns the prefix of a given infinite sequence while the second returns the suffix. Both are easily defined by induction on the first argument.

The issues of (1) and of (2) arise in the verification of the sheaf condition. We mentioned in Chapter 3.3.1 that the following sheaf conditions are equivalent:

(s1) For any $n \in \mathbb{N}$ and any family $\{p_s \in P\}_{s\in\mathbf{2}^n}$, the map $p\colon \mathbf{2}^{\mathbb{N}} \to X$ defined by $p(s\alpha) = p_s(\alpha)$ is in P.

(s2) For any $p_0, p_1 \in P$, the map $p\colon \mathbf{2}^{\mathbb{N}} \to X$ defined by $p(i\alpha) = p_i(\alpha)$ is in P.

(s3) If $p\colon \mathbf{2}^{\mathbb{N}} \to X$ a map such that there exists $n \in \mathbb{N}$ with $p \circ \mathrm{cons}_s \in P$ for all $s \in \mathbf{2}^n$, then $p \in P$.

Here X is a set and P is the desired C-topology on X. Under the Curry–Howard interpretation, they become the following types:

(s1) $\Pi(n{:}\mathbb{N}).\ \Pi(\rho\colon \mathbf{2}^n \to \mathbf{2}^{\mathbb{N}} \to X).\ (\Pi(s{:}\mathbf{2}^n).\ \rho_s \in P) \to \mathrm{Aml}_n(\rho) \in P$,

(s2) $\Pi(\rho\colon \mathbf{2} \to \mathbf{2}^{\mathbb{N}} \to X).\ (\Pi(i{:}\mathbf{2}).\ \rho_i \in P) \to \mathrm{Aml}_{\mathbf{2}}(\rho) \in P$,

(s3) $\Pi(p\colon \mathbf{2}^{\mathbb{N}} \to X).\ (\Sigma(n{:}\mathbb{N}).\ \Pi(s{:}\mathbf{2}^n).\ p \circ \mathrm{cons}_s \in P) \to p \in P$,

where X is a type, P is a family $(\mathbf{2}^{\mathbb{N}} \to X) \to \mathcal{U}$, the map $\mathrm{Aml}_n(\rho)\colon \mathbf{2}^{\mathbb{N}} \to X$ is defined by $\mathrm{Aml}_n(\rho)(\alpha) :\equiv \rho\ (\mathrm{take}\ n\ \alpha)\ (\mathrm{drop}\ n\ \alpha)$, and the map $\mathrm{Aml}_{\mathbf{2}}(\rho)\colon \mathbf{2}^{\mathbb{N}} \to X$ is defined by $\mathrm{Aml}_{\mathbf{2}}(\rho)(\alpha) :\equiv \rho_{\alpha_0}(\alpha \circ \mathrm{succ})$. However, the above types are *not* equivalent unless (funext) is available. For example, we need $\mathrm{Aml}_n(\rho) \circ \mathrm{cons}_s = \rho_s$ to prove the equivalence of (s1) and (s3) which clearly requires (funext). In particular, we cannot work with (s2) because verifying it for exponentials requires the equivalence with (s1) or (s3) (see the proof of Theorem 3.3.4). We find (s3) more convenient to work with, *e.g.* to cooperate with the coverage axiom since both involve cons maps. Working with the sheaf condition (s3), we notice that the constructions of (1) and of (2) rely on on the equation

$$\mathrm{cons}\ (\mathrm{take}\ n\ \alpha)\ (\mathrm{drop}\ n\ \alpha) = \alpha$$

for all $n\colon \mathbb{N}$ and $\alpha\colon \mathbf{2}^{\mathbb{N}}$, which holds up to *pointwise* equality only, *i.e.* we can only prove $\Pi(i{:}\mathbb{N}).\ (\mathrm{cons}\ (\mathrm{take}\ n\ \alpha)\ (\mathrm{drop}\ n\ \alpha))_i = \alpha_i$.

In case (1) regarding discrete C-spaces, we attempt to prove the sheaf condition (s3) as follows. Given a type X, the discrete C-topology is a type family $\mathrm{LC}\colon (\mathbf{2}^{\mathbb{N}} \to X) \to \mathcal{U}$ defined by

$$\mathrm{LC}(p) :\equiv \Sigma(m{:}\mathbb{N}).\ \Pi(\alpha, \beta : \mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to p(\alpha) = p(\beta)$$

which expresses that the map p is locally constant. Let a map $p\colon \mathbf{2}^{\mathbb{N}} \to X$, a natural number $n\colon \mathbb{N}$, and a proof of $\Pi(s{:}\mathbf{2}^n).\ p \circ \mathrm{cons}_s \in \mathrm{LC}$ be given. The goal is to prove $p \in \mathrm{LC}$. Let m_s be the modulus of local constancy of $p \circ \mathrm{cons}_s$ for each $s\colon \mathbf{2}^n$, and let m

be the maximum of all m_s's. We want to prove that $n + m$ is a modulus of p: given $\alpha, \beta \colon \mathbf{2}^{\mathbb{N}}$ with $\alpha =_{n+m} \beta$, we have

$$\text{take } n\ \alpha = \text{take } n\ \beta \qquad \text{and} \qquad \text{drop } n\ \alpha =_m \text{drop } n\ \beta.$$

Let $s :\equiv \text{take } n\ \alpha$. By the local constancy of $p \circ \text{cons}_s$ we have

$$p(\text{cons}_s(\text{drop } n\ \alpha)) = p(\text{cons}_s(\text{drop } n\ \beta)).$$

But this is insufficient to conclude $p\alpha = p\beta$ because the equation $\text{cons}_s(\text{drop } n\ \alpha) = \alpha$ holds up to pointwise equality as discussed above.

In case (2) regarding exponentials of C-spaces, the verification of (any of the above formulations of) the sheaf condition uses the coverage axiom, which is formulated as the following type:

$$\Pi(t \colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}).\ t \in \mathrm{C} \to$$
$$\Pi(m{:}\mathbb{N}).\ \Sigma(n{:}\mathbb{N}).\ \Pi(s{:}\mathbf{2}^n).\ \Sigma(s'{:}\mathbf{2}^m).\ \Sigma(t' \colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}).\ t' \in \mathrm{C} \times t \circ \text{cons}_s = \text{cons}_{s'} \circ t'.$$

To prove that the coverage axiom holds, we use one direction of Lemma 3.2.1: given a uniformly continuous map t and a natural number m, we get n using the uniform-continuity witness of t; given $s \colon \mathbf{2}^n$, we construct $s' \colon \mathbf{2}^m$ and $t' \colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ by taking

$$s' :\equiv \text{take } m\ (t(\text{cons}_s 0^\omega)) \qquad \text{and} \qquad t'(\alpha) :\equiv \text{drop } m\ (t(\text{cons}_s \alpha))$$

where $0^\omega \colon \mathbf{2}^{\mathbb{N}}$ is the constantly zero sequence, *i.e.* $(0^\omega)_i :\equiv 0$ for all $i \colon \mathbb{N}$. We can easily show $t' \in \mathrm{C}$. However, we are not able to prove $t \circ \text{cons}_s = \text{cons}_{s'} \circ t'$ or its weakening $\Pi(\alpha{:}\mathbf{2}^{\mathbb{N}}).\ t(\text{cons}_s \alpha) = \text{cons}_{s'}(t'\alpha)$, as the latter equation can be expanded to

$$t(\text{cons}_s \alpha) = \text{cons}\ (\text{take } m\ (t(\text{cons}_s \alpha)))\ (\text{drop } m\ (t(\text{cons}_s \alpha)))$$

which holds up to pointwise equality as mentioned above. Therefore, without (funext) we can only prove

$$\Pi(\alpha{:}\mathbf{2}^{\mathbb{N}}).\ \Pi(i{:}\mathbb{N}).\ (t(\text{cons}_s \alpha))_i = (\text{cons}_{s'}(t'\alpha))_i$$

for the coverage axiom, which is clearly insufficient for verifying the sheaf condition for exponentials as $t \circ \text{cons}_s = \text{cons}_{s'} \circ t'$ is required.

There are two issues in the construction and proof of (3), regarding (3a) the discreteness of the exponential $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$ and (3b) the desired property of the fan functional. Both of them are related to the lack of (funext) and the presence of proof relevance:

(3a) To show the discreteness of $\mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$, *i.e.* to show that any given probe $p \colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}^{\mathbf{2}^{\mathbb{N}}}$ is locally constant, we follow the proof of Lemma 3.5.1, and get a natural number n which we want to prove to be a modulus of local constancy of p, *i.e.* given any $\alpha, \beta \colon \mathbf{2}^{\mathbb{N}}$ with $\alpha =_n \beta$, we want to prove $p\alpha = p\beta$. Since $p\alpha$ and $p\beta$ are continuous maps from the exponential $\mathbf{2}^{\mathbb{N}}$ to the discrete C-space $\mathbb{N}$, we need to show both equalities of their underlying maps and of their continuity witnesses. In the proof of Lemma 3.5.1, we prove the equality of the underlying maps using (funext) whose usage seems to be unavoidable. However, there is *no* way to prove that their continuity witnesses are also equal: the continuity of a map $f \colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ means that

$f \circ q \colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ is uniformly continuous (or locally constant) whenever $q \colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ is a probe on the exponential $\mathbf{2}^{\mathbb{N}}$; but even for the same uniformly continuous map, it can have more than one witnesses of uniform continuity because if n is a modulus then so is any number greater than n.

(3b) In order to validate the uniform-continuity principle, we have to show that the fan functional computes moduli of uniform continuity of continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$, *i.e.* for any continuous maps $f \colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ and $\alpha, \beta \colon \mathbb{N} \to \mathbf{2}$ (all of them are pairs consisting of a function and a continuity witness), the type

$$\mathrm{pr}_1(\alpha) =_{\mathrm{pr}_1(\mathrm{fan})(f)} \mathrm{pr}_1(\beta) \;\to\; \mathrm{pr}_1(f)(\alpha) = \mathrm{pr}_1(f)(\beta)$$

is inhabited. In the proof of Theorem 3.5.2, we show that any given continuous map $f \colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ is uniformly continuous, because the identity map 1 is a probe on $\mathbf{2}^{\mathbb{N}}$ and thus $f = f \circ 1$ is a probe on $\mathbb{N}$. However, in a proof-relevant setting, the identity map is not a probe on the exponential $\mathbf{2}^{\mathbb{N}}$, because an element of the exponential $\mathbf{2}^{\mathbb{N}}$ is a pair consisting of a map $\mathbb{N} \to \mathbf{2}$ and a continuity witness. Thus we have to explicitly defined an "identity" probe $\mathrm{I} \colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ which equips a sequence $\mathbb{N} \to \mathbf{2}$ with a continuity witness using the discreteness of the space $\mathbf{2}$, to make it an element of the exponential $\mathbf{2}^{\mathbb{N}}$. Now, if $\mathrm{pr}_1(\alpha) =_{\mathrm{pr}_1(\mathrm{fan})(f)} \mathrm{pr}_1(\beta)$ is known, what we can prove is $\mathrm{pr}_1(f)(\mathrm{I}(\mathrm{pr}_1\alpha)) = \mathrm{pr}_1(f)(\mathrm{I}(\mathrm{pr}_1\beta))$. Therefore, to get the desired result, we need $\mathrm{I}(\mathrm{pr}_1\alpha) = \alpha$ for all continuous maps $\alpha \colon \mathbb{N} \to \mathbf{2}$. This requires to prove equality of continuity witnesses of a given map $\mathbb{N} \to \mathbf{2}$, which is impossible without further adjustments as discussed in (3a).

Both issues are related to the fact that uniform continuity is not a proposition, *i.e.* a map $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ (or $\mathbf{2}^{\mathbb{N}} \to \mathbf{2}$) can have more than one witnesses of uniform continuity. To solve such issues, we refine uniform continuity to mean that there exists a *minimal* modulus of uniform continuity. Hence, given a map $f \colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$, its uniform continuity (or local constancy) is formulated as

$$\mathrm{UC}_{\mathbb{N}}(f) :\equiv \Sigma_{\min}(n{:}\mathbb{N}).\ \Pi(\alpha, \beta : \mathbf{2}^{\mathbb{N}}).\ \alpha =_n \beta \to f\alpha = f\beta,$$

where $\Sigma_{\min}$ is defined by, for any family $P \colon \mathbb{N} \to \mathcal{U}$

$$\Sigma_{\min}(n{:}\mathbb{N}).P(n) :\equiv \Sigma(n{:}\mathbb{N}).\ P(n) \times (\Pi(m{:}\mathbb{N}).\ P(m) \to n \leq m).$$

Now a uniform-continuity witness of f becomes a tuple, consisting of a natural number n, a proof that n is a modulus of uniform continuity of f, and a proof that any modulus of uniform continuity of f is greater than or equal to n. Given any two uniform-continuity witnesses, the minimal moduli (*i.e.* the first components) must be the same. Then, using (funext) and the fact that the types $n = m$ and $n \leq m$ are propositions for any $n, m : \mathbb{N}$, one can show that the above refinement of uniform continuity is a proposition. Moreover, if two uniformly continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are pointwise equal, then they have the same minimal modulus and thus yield the same morphism in the category of C-spaces, which solves issue (3a). We similarly refine uniform continuity of maps $f \colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}$ to

$$\mathrm{UC}_{\mathbf{2}}(f) :\equiv \Sigma_{\min}(n{:}\mathbb{N}).\ \Pi(\alpha, \beta : \mathbf{2}^{\mathbb{N}}).\ \alpha =_n \beta \to f\alpha = f\beta$$

in order to tackle issue (3b). Another benefit of this refinement is that the minimality of moduli is maintained during the construction of the model. Thus the definition of the fan functional does not need to additionally compute the minimal modulus from a given one as in the proof of Theorem 3.5.2. However, with this refinement, we still need (funext) to complete the type-theoretic implementation of the fan functional.

7.2 Construction via different approaches

To address the issues caused by the lack of (funext), we developed the following approaches, all of which have been implemented in Agda and are available at [76].

Postulate (funext). This is probably the simplest approach which produces the most readable and clean type-theoretic implementation of our work. We present the main definitions and constructions in Section 7.2.1. The drawback of this approach is that it does not compute in Agda. We give explicit examples in Section 7.2.1 and explain why they can or cannot be normalized to numerals. (The Agda implementation can be easily, but laboriously, translated to the cubical type theory by Coquand *et al.* [9, 19], which does compute in the presence of funext. But this is left for future work.)

Use setoids. In this approach, we work with *setoids* [41], that is, with types equipped with equivalence relations. We discuss, in Section 7.2.2, the main adjustments to the model construction and explain how they solve the above issues. With this approach, we successfully implemented the model in intensional MLTT, which allows us to compute least moduli of uniform continuity of T-definable functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$. But one of its drawbacks, as usual, is that it makes the constructions and proofs tedious, long and unreadable.

Add a probe axiom. This was motivated by an important adjustment of the approach based on setoids. We add the following probe axiom

> If a map $\mathbf{2}^{\mathbb{N}} \to X$ is pointwise equal to a probe on X, then it is a probe on X.

The intuition is that this axiom is playing the role of the "transport" function. As discussed in Section 7.2.2, it avoids some usages of (funext), *e.g.* in the proof that any map from a discrete C-space is continuous, and hence preserves some amount of computational content, *e.g.* of the examples in our experiment of computing moduli of uniform continuity. However, (funext) is still required in some other constructions and proofs.

Postulate (funext) within a computationally irrelevant field. In earlier stages of this work, we conjectured that (funext) occurs only in computationally irrelevant contexts. In order to attempt to verify it, we made use of Agda's *irrelevant fields* and postulated (funext) within such an irrelevant context [1]. In fact, the Agda type checker proved our conjecture false. To make this idea work, we added a probe axiom similar to (actually stronger than) the one of the previous approach. The first drawback of this approach is that it produces a different model (or category) from the original one as introduced in Chapter 3.3 because of the additional probe axiom. Another drawback is that it requires the extension of type theory with irrelevant fields. We will not discuss this approach further in the thesis. But the corresponding Agda implementation is available at [76].

Postulate the double negation of (funext). This was based on the observation that the only property of irrelevant fields we used is that they form a monad T satisfying

$T\emptyset \to \emptyset$. Since double negation is the final such monad, *i.e.* $TX \to \neg\neg X$, we can instead postulate $\neg\neg$(funext). We present the main adjustments of this approach, one of which is to add a probe axiom as in the previous approaches, in Section 7.2.4. From a constructive point of view, the model produced using this approach is not the same as the original one. However, they provide the same interpretation to simple (and to dependent) types. And more interestingly, it does not destroy computational content because the consistent axiom we postulated is in a negative form [22].

7.2.1 Construction by postulating (funext)

In this section we present the main construction of the model in intensional MLTT extended with the axiom of function extensionality, which we have formalized in Agda notation by postulating the axiom (funext) [76]. However, the Agda implementation fails to compute moduli of uniform continuity, because the postulated term (funext) has no computation rules. In the end, we provide two examples of moduli of uniform continuity and explain why they can and cannot be normalized to numerals.

Chapter 3.3.1 defines a C-topology P on a set X to be a collection of maps $\mathbf{2}^{\mathbb{N}} \to X$ satisfying the following probe axioms:

(1) All constant maps are in P.

(2) If $p \in P$ and $t \in \mathrm{C}$, then $p \circ t \in P$.

(3) If $p\colon \mathbf{2}^{\mathbb{N}} \to X$ is a map such that there exists $n \in \mathbb{N}$ with $p \circ \mathrm{cons}_s \in P$ for all $s \in \mathbf{2}^n$, then $p \in P$.

Given $X : \mathcal{U}$ and $P\colon (\mathbf{2}^{\mathbb{N}} \to X) \to \mathcal{U}$, these probe axioms are translated to the following product type, written as probe-axioms(X, P), under the Curry–Howard interpretation

$$\begin{aligned} & (\Pi(x{:}X).\ \lambda\alpha.x \in P) \\ \times\ & \big(\Pi(p\colon \mathbf{2}^{\mathbb{N}} \to X).\ p \in P \to \Pi(t\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}).\ t \in \mathrm{C} \to p \circ t \in P\big) \\ \times\ & \big(\Pi(p\colon \mathbf{2}^{\mathbb{N}} \to X).\ (\Sigma(n{:}\mathbb{N}).\ \Pi(s{:}\mathbf{2}^n).\ p \circ \mathrm{cons}_s \in P) \to p \in P\big)\,. \end{aligned}$$

Remark. A naive formulation of the first probe axiom is

$$\Pi(p\colon \mathbf{2}^{\mathbb{N}} \to X).\ (\Pi(\alpha, \beta\colon \mathbf{2}^{\mathbb{N}}).\ p\alpha = p\beta) \to p \in P$$

which is logically equivalent to $\Pi(x{:}X).\ \lambda\alpha.x \in P$ under the assumption of (funext). The choice of these two formulations affects the proofs of the following:

(a) The chosen type family for exponentials satisfies the first probe axiom.

(b) All maps from discrete C-spaces are continuous.

When working with the longer formulation, the proof of (a) needs to use (funext) while the one of (b) does not. However, with the shorter formulation we choose, the proof of (a) does not require (funext) but the one of (b) does. Therefore, working with any of them cannot avoid the usage of (funext) without further adjustments. More details will be provided when discussing exponentials and discrete C-spaces below.

The type of C-spaces is then formulated as

$$\text{Space} :\equiv \Sigma(X{:}\mathcal{U}).\ \Sigma(P\colon (\mathbf{2}^{\mathbb{N}} \to X) \to \mathcal{U}).\ \text{probe-axioms}(X, P).$$

In words, an element $X\colon \text{Space}$ consists of a underlying set/type $|X|\colon \mathcal{U}$, a type family $\text{Probe}(X)\colon (\mathbf{2}^{\mathbb{N}} \to |X|) \to \mathcal{U}$ representing the C-topology on $|X|$, and a proof of the probe axioms. The operators $|-|\colon \text{Space} \to \mathcal{U}$ and $\text{Probe}\colon \Pi(X{:}\text{Space}).\ (\mathbf{2}^{\mathbb{N}} \to |X|) \to \mathcal{U}$ are easily defined using the projections pr_1 and pr_2. Given $X, Y\colon \text{Space}$, we write, for any map $f\colon |X| \to |Y|$,

$$\text{continuous}(f) :\equiv \Pi(p\colon \mathbf{2}^{\mathbb{N}} \to |X|).\ p \in \text{Probe}(X) \to f \circ p \in \text{Probe}(Y)$$

and then define the type of continuous maps of X and Y by

$$\text{Map}(X, Y) :\equiv \Sigma(f\colon |X| \to |Y|).\ \text{continuous}(f).$$

A continuous map is thus a pair consisting of a underlying map and a continuity witness.

For the cartesian closed structure, we present only the constructions of underlying sets and of type families (C-topologies), but omit the verifications of probe axioms and proofs of universal properties:

1. The *terminal* C-space is (proved to be) the singleton type $\mathbf{1}$ equipped with the constant family $(\mathbf{2}^{\mathbb{N}} \to \mathbf{1}) \to \mathcal{U}$ which sends the (unique) map $\mathbf{2}^{\mathbb{N}} \to \mathbf{1}$ to type $\mathbf{1}$.

2. Given C-spaces X and Y, the *product* $X \times Y$ consists of the binary product type $|X| \times |Y|$ as its underlying set and the family $R\colon (\mathbf{2}^{\mathbb{N}} \to |X| \times |Y|) \to \mathcal{U}$ defined by

   $$r \in R :\equiv \text{pr}_1 \circ r \in \text{Probe}(X) \times \text{pr}_2 \circ r \in \text{Probe}(Y)$$

   as its C-topology.

3. Given C-spaces X and Y, the *exponential* Y^X consists of the type $\text{Map}(X, Y)$ of continuous maps as its underlying set and the family $R\colon (\mathbf{2}^{\mathbb{N}} \to \text{Map}(X, Y)) \to \mathcal{U}$ defined by

   $$\begin{aligned} r \in R :\equiv\ & \Pi(p\colon \mathbf{2}^{\mathbb{N}} \to |X|).\ p \in \text{Probe}(X) \to \\ & \Pi(t\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}).\ t \in \text{C} \to \lambda\alpha.\text{pr}_1(r(t\alpha))(p\alpha) \in \text{Probe}(Y) \end{aligned}$$

   as its C-topology.

   As mentioned above, verifying the shorter formulation of the first probe axiom is easy, because when r is $\lambda\alpha.f$ for any $f\colon \text{Map}(X, Y)$ we have $\lambda\alpha.\text{pr}_1(r(t\alpha))(p\alpha) = \text{pr}_1(f) \circ p$ which is a probe on Y according to the continuity witness $\text{pr}_2(f)$.

In order to solve the issues related to proof relevance (see Section 7.1), the discrete C-topology $\text{UC}_{\mathbb{N}}\colon (\mathbf{2}^{\mathbb{N}} \to \mathbb{N}) \to \mathcal{U}$ on natural numbers is refined as

$$p \in \text{UC}_{\mathbb{N}} :\equiv \Sigma_{\min}(n{:}\mathbb{N}).\ \Pi(\alpha, \beta : \mathbf{2}^{\mathbb{N}}).\ \alpha =_n \beta \to p\alpha = p\beta,$$

and the one $\text{UC}_{\mathbf{2}}\colon (\mathbf{2}^{\mathbb{N}} \to \mathbf{2}) \to \mathcal{U}$ on booleans is refined as

$$p \in \text{UC}_{\mathbf{2}} :\equiv \Sigma_{\min}(n{:}\mathbb{N}).\ \Pi(\alpha, \beta : \mathbf{2}^{\mathbb{N}}).\ \alpha =_n \beta \to p\alpha = p\beta.$$

For the sake of readability, we write $\mathbb{N}$ to mean the discrete C-space of natural numbers, $\mathbf{2}$ to mean the discrete two-point C-space, and $\mathbf{2}^{\mathbb{N}}$ to mean the exponential of $\mathbf{2}$ to the power $\mathbb{N}$, by abuses of notations.

One direction of Lemma 3.3.8 is that if the probes of a C-space X are precisely the locally constant maps then X is discrete (all maps from X are continuous). It is useful, for instance, for proving the continuity of the recursion combinator of natural numbers. As mentioned earlier, we need (funext) to prove it because we are working with the shorter formulation of the first probe axiom: Let a map $f\colon |X| \to |Y|$ be given. If $p\colon \mathbf{2}^{\mathbb{N}} \to |X|$ is locally constant with modulus n, then for any $s \in \mathbf{2}^n$ the map $f \circ p \circ \mathrm{cons}_s$ is constant. By the first probe axiom, the map $\lambda\alpha.f(p(\mathrm{cons}_s 0^\omega))$ is in $\mathrm{Probe}(Y)$. Using (funext) we have $f \circ p \circ \mathrm{cons}_s = \lambda\alpha.f(p(\mathrm{cons}_s 0^\omega))$ and thus $f \circ p \circ \mathrm{cons}_s \in \mathrm{Probe}(Y)$.

As mentioned in Section 7.1, we constructed an "identity" probe

$$\mathrm{id}\colon \mathbf{2}^{\mathbb{N}} \to \Sigma(\alpha\colon \mathbb{N} \to \mathbf{2}).\ \mathrm{continuous}(\alpha)$$

of the exponential $\mathbf{2}^{\mathbb{N}}$, the first component of whose output is the same as the input, and the second component is constructed by using the lemma that all maps from $\mathbb{N}$ are continuous. In other words, id equips any map $\mathbb{N} \to \mathbf{2}$ with a continuity witness to make it a continuous map. The following direction of the Yoneda Lemma 3.4.2

$$\Pi(X\colon \mathrm{Space}).\ \mathrm{Map}(\mathbf{2}^{\mathbb{N}}, X) \to \Sigma(p\colon \mathbf{2}^{\mathbb{N}} \to |X|).\ p \in \mathrm{Probe}(X)$$

is proved by composing a given continuous map with the above "identity" probe. When X is the discrete C-space $\mathbb{N}$, for any $f\colon \mathrm{Map}(\mathbf{2}^{\mathbb{N}}, \mathbb{N})$, we have a map $\mathrm{pr}_1(f) \circ \mathrm{id}\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ which is a probe on $\mathbb{N}$, *i.e.* a uniformly continuous map. Using the uniform-continuity witness $\mathrm{pr}_2(f)$, we get the least modulus m of uniform continuity of $\mathrm{pr}_1(f) \circ \mathrm{id}$. Since the probe id does not change the input sequences, we can show that m is also the least modulus of uniform continuity of f in the following sense:

$$\Pi(\alpha, \beta\colon \mathrm{Map}(\mathbb{N}, \mathbf{2})).\ \mathrm{pr}_1(\alpha) =_m \mathrm{pr}_1(\beta) \to \mathrm{pr}_1(f)(\alpha) = \mathrm{pr}_1(f)(\beta).$$

This is how we construct the underlying map of the fan functional. We omit the proof of its continuity which uses (funext) as discussed in Section 7.1.

Since (funext) is a constant without computational rule, the computational content of our formal development is destroyed. For example, we can get a closed term of type $\mathbb{N}$ as follows: Define a closed T-term F of type $(\mathsf{N} \to 2) \to \mathsf{N}$, and then apply the interpretation to get a continuous map $[\![F]\!]\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$. (The inductive definition of system T and its interpretation function into C-spaces can be found in Chapter 8.2 in Agda notation.) Using the fan functional, we get its least modulus of uniform continuity, *i.e.* a closed term $\mathrm{pr}_1(\mathrm{fan})([\![F]\!])\colon \mathbb{N}$. However, not all such closed terms can be normalized to numerals:

1. The following closed T-term

   $$F_0 :\equiv \lambda\alpha.0 : (\mathsf{N} \to 2) \to \mathsf{N}$$

   is interpreted as a constantly zero function. Recall that the continuity witness of $[\![0]\!]$ is a constant map sending probes on $\mathbf{2}^{\mathbb{N}}$ to the uniform-continuity witness of $\lambda\alpha.0$ whose first component (the least modulus of uniform continuity) is clearly 0.

Hence, Agda normalizes $\mathrm{pr}_1(\mathrm{fan})(\llbracket F_0 \rrbracket)$ to 0, and in this case we have no loss of computational content.

2. However, our Agda implementation fails to compute the (least) modulus of uniform continuity of the following closed T-term

$$F_1 :\equiv \lambda\alpha.\ \mathsf{if}\ \mathsf{t}\ 0\ 0 : (\mathsf{N} \to 2) \to \mathsf{N}$$

which is also interpreted as a constantly zero function. Because the continuity witness of $\llbracket \mathsf{if} \rrbracket$ is constructed using Lemma 3.3.8 whose proof uses (funext) as discussed before, the closed term $\mathrm{pr}_1(\mathrm{fan})(\llbracket F_1 \rrbracket)$ contains (funext) which stops the normalization as (funext) has no computational rules. The normal form of $\mathrm{pr}_1(\mathrm{fan})(\llbracket F_1 \rrbracket)$ produced by Agda has more than 300 lines in the way Agda outputs it [76, module `UsingFunext.ModellingUC.ComputationExperiments`].

Remark. As explained above, the fan functional computes the least modulus of uniform continuity of a map $f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ using only the continuity witness of f without looking into its underlying map. Thus, though $\llbracket F_0 \rrbracket$ and $\llbracket F_1 \rrbracket$ have the same underlying map, Agda successfully computes the least modulus of uniform continuity of F_0, but fails to compute the one of F_1. One can even prove that the two continuous maps $\llbracket F_0 \rrbracket$ and $\llbracket F_1 \rrbracket$ are identical. However, this does not help the computation because again this proof unavoidably uses (funext).

7.2.2 Construction by using setoids

Recall that the type of setoids in MLTT is formulated as

$$\mathrm{Setoid} :\equiv \Sigma(X\colon \mathcal{U}).\ \Sigma(R\colon X \to X \to \mathcal{U}).\ \mathrm{isEqRel}(R)$$

where $\mathrm{isEqRel}(R)$ is the following product type

$$\begin{array}{ll} & (\Pi(x\colon X).\ R(x,x)) \\ \times & (\Pi(x,y\colon X).\ R(x,y) \to R(y,x)) \\ \times & (\Pi(x,y,z\colon X).\ R(x,y) \to R(y,z) \to R(x,z)) \end{array}$$

expressing that R is reflexive, symmetric and transitive. The type of morphisms, also called *extensional maps*, of setoids X and Y is formulated as

$$\text{E-map}(X,Y) :\equiv \Sigma(f\colon |X| \to |Y|).\ \Pi(x,x'\colon |X|).\ x \sim_X x' \to fx \sim_Y fx'$$

where $|X|$ is the underlying type of X and $\sim_X$ is the equivalence relation on $|X|$.

Compared to the development in Section 7.2.1, several adjustments need to be applied to the model constructions in order to solve the issues discussed in Section 7.1:

1. For the notion of uniform continuity, we consider only the extensional endomaps of $\mathbf{2}^{\mathbb{N}}$, *i.e.* a uniformly continuous map $t\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ should also satisfy

$$\Pi(\alpha,\beta\colon \mathbf{2}^{\mathbb{N}}).\ (\Pi(i{:}\mathbb{N}).\alpha_i = \beta_i) \to (\Pi(i{:}\mathbb{N}).(t\alpha)_i = (t\beta)_i).$$

This clearly solves issues of (1) and of (2), because now only pointwise equality is required for elements of $\mathbf{2}^{\mathbb{N}}$.

2. Given $X\colon$ Setoid, we consider only the extensional maps $\mathbf{2}^{\mathbb{N}} \to |X|$ to be probes, *i.e.* a C-topology on X is a type family $P\colon \text{E-map}(\mathbf{2}^{\mathbb{N}}, X) \to \mathcal{U}$ satisfying the probe axioms. Recall that any predicate P on a setoid X should preserve its equivalence relation, *i.e.* $\Pi(x, y\colon X).\ x \sim_X y \to P(x) \to P(y)$. Therefore, we have to require the C-topology P to additionally satisfy

$$\Pi(p, q\colon \text{E-map}(\mathbf{2}^{\mathbb{N}}, X)).\ (\Pi(\alpha\colon \mathbf{2}^{\mathbb{N}}).\mathrm{pr}_1(p)(\alpha) \sim_X \mathrm{pr}_1(q)(\alpha)) \to p \in P \to q \in P.$$

   In words, if a map q is pointwise equal to a probe p then q is also a probe. This additional condition avoids transporting probe-witnesses along equality of maps obtained using (funext).

3. Continuous maps of C-spaces now have to be extensional. The collection of continuous, extensional maps of C-spaces X and Y, together with the equivalence relation defined pointwise, forms the underlying setoid of the exponential Y^X. This defines a notion of equality of continuous maps that ignores continuity witnesses, which solves the issues of (3) with no need of requiring minimal moduli of uniform continuity. But we remark that, when defining the fan functional, we have to calculate minimal moduli of given continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ as in the proof of Theorem 3.5.2, because the fan functional has to be extensional, *i.e.* for any two continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ which are pointwise equal, the fan functional should give the same modulus of uniform continuity.

This approach allows us to implement the model without postulating any non-standard axiom for intensional MLTT. However, it gives a longer and less readable formalization that obscures the essential aspects of the constructions and proofs. Hence we looked for an approach that gives a concise formalization without losing computational content.

7.2.3 Construction by adding an probe axiom

Motivated by the second adjustment in the approach of using setoids (Section 7.2.2), we add the following probe axiom:

(4) If a map is pointwise equal to a probe, then it is a probe.

This is formulated as

$$\Pi(p, q\colon \mathbf{2}^{\mathbb{N}} \to X).\ p \in P \to (\Pi(\alpha\colon \mathbf{2}^{\mathbb{N}}).p\alpha = q\alpha) \to q \in P$$

where $X\colon \mathcal{U}$ is the underlying type/set of a C-space, and $P\colon (\mathbf{2}^{\mathbb{N}} \to X) \to \mathcal{U}$ is the C-topology on X.

Now, in each case the construction of a C-space is performed, we have to verify the above axiom. For example, the discrete C-space $\mathbb{N}$ of natural numbers satisfies the axiom, because if a map p is uniformly continuous and q is pointwise equal to p then clearly q is also uniformly continuous (with the same least modulus as of p). Moreover, for any two C-spaces X and Y, our previous constructions of product $X \times Y$, of coproduct $X + Y$ and of exponential X^Y satisfy the above probe axiom.

As mentioned in Section 7.2.2, this additional probe axiom is playing the role of the transport function. More specifically, it avoids transporting probe witnesses along paths constructed using (funext) in the proofs of the following:

1. *Exponentials satisfy the sheaf condition.* Recall that the proof of this fact uses the coverage axiom. As discussed in Section 7.1, without using (funext), the required equation in the coverage axiom holds only up to pointwise equality, *i.e.*

$$\Pi(\alpha{:}\mathbf{2}^{\mathbb{N}}).\ \Pi(i{:}\mathbb{N}).\ (t(\mathrm{cons}_s\alpha))_i = (\mathrm{cons}_{s'}(t'\alpha))_i.$$

   With probe axiom (4), rather than $t \circ \mathrm{cons}_s = \mathrm{cons}_{s'} \circ t'$, the pointwise equality

$$\Pi(\alpha{:}\mathbf{2}^{\mathbb{N}}).\ t(\mathrm{cons}_s\alpha) = \mathrm{cons}_{s'}(t'\alpha)$$

   is sufficient to verify the sheaf condition for exponentials. However, (funext) is still needed to obtain the above pointwise equality.

2. *Any map from a discrete* C*-space is continuous.* In Section 7.2.1 we prove this fact by using (funext): we know that a map is a probe and that another map is pointwise equal to it, and then conclude that the second map is also a probe by applying (funext) to get an equality of the two maps. It is clear that, with probe axiom (4), the usage of (funext) here is no longer needed.

In Section 7.2.1, the least modulus of uniform continuity of the T-function $\lambda\alpha.$ `if t 0 0` cannot be normalized to a numeral. By adding the probe axiom (4), the continuity witness of the case-distinction function `if` is constructed without using (funext). Therefore, Agda successfully normalizes its least modulus of uniform continuity to the numeral 0. Moreover, all the examples in our experiments, reported in Chapter 8.3, can be successfully normalized to numerals. However, because (funext) is still needed, *e.g.* in the verification of sheaf condition for exponentials and in the continuity witness of the fan functional, the computational content of our development is still potentially destroyed.

7.2.4 Construction by postulating ¬¬(funext)

After introducing an additional probe axiom, as discussed in Section 7.2.3, the least uniform-continuity moduli of all the T-definable functions in our experiment (Chapter 8.3) can be normalized to numerals. Hence, we conjectured that the usages of (funext) had no computational content, and made use of Agda's *irrelevant fields* [1] to attempt to verify this idea. We had two observations in the approach of using irrelevant fields:

1. We actually need a condition that is stronger than probe axiom (4), and hence get a model which is not the same as the one introduced in the previous chapters.

2. The only property of irrelevant fields we used to construct the model is that they form a monad T satisfying $T\emptyset \to \emptyset$.

It is well known that double negation ¬¬ is the *final* monad T such that $T\emptyset \to \emptyset$. Hence we instead postulated the double negation of (funext) and added the following stronger form of probe axiom (4)

$$\Pi(p, q{:}\ \mathbf{2}^{\mathbb{N}} \to X).\ p \in P \to (\Pi(\alpha{:}\ \mathbf{2}^{\mathbb{N}}).\ \neg\neg(p\alpha = q\alpha)) \to q \in P.$$

Here ¬¬ cannot be removed because, in the verification of the sheaf condition for exponentials, we have two pointwise equal sequences $\alpha, \beta{:}\ \mathbf{2}^{\mathbb{N}}$ (given by the pointwise coverage

axiom explained in Section 7.1), and using $\neg\neg$(funext) we can only conclude $\neg\neg(p\alpha = p\beta)$ for some map p.

One can easily prove that the above probe axiom is preserved by the constructions of products, coproducts and exponentials. To show that the discrete space $\mathbb{N}$ (and $\mathbf{2}$) satisfies the probe axiom, we use the following facts:

(a) $\neg\neg A \to A$ holds for any decidable type A, *i.e.*

$$\Pi(A\colon \mathcal{U}).\ (A + \neg A) \to \neg\neg A \to A$$

which can be proved by case analysis on $A + \neg A$.

(b) Natural numbers have decidable equality, *i.e.*

$$\Pi(n, m\colon \mathbb{N}).\ (n = m) + \neg(n = m).$$

which can be proved by induction on both n and m.

We remark that the above facts are also needed when proving the continuity of the fan functional.

There are a few advantages of this approach:

- It provides a relatively concise development of the work in type theory.
- It does not require non-standard extension of MLTT such as irrelevant fields.
- With classical logic, we get the same model as the one explained in the previous chapters.

  Every type becomes decidable under classical logic; thus, the usage of $\neg\neg$ can be got rid of in the development.

- Using (funext) we can prove that this model gives an equivalent interpretation to simple (and to dependent) types as the original one.

  This can be verified by induction on the (simple and dependent) type structure. A type A is said to be $\neg\neg$-*separated* if $\Pi(x, y\colon A).\ \neg\neg(x = y) \to x = y$. We know $\mathbb{N}$ is $\neg\neg$-separated as discussed above. And (funext) is needed when showing that the function space $A \to B$ is $\neg\neg$-separated whenever B is $\neg\neg$-separated. Therefore, the underlying type of any simple C-spaces is $\neg\neg$-separated.

- More interestingly and more importantly, it does not destroy the computational content of the development.

  This is because the canonicity of a type theory is preserved when a negative consistent axiom is postulated. More detailed discussion of this meta-theorem and some examples of using it are provided in [22]. Here we briefly recall its proof: Consider a consistent closed type $\neg A$ and extend MLTT with a constant $c\colon \neg A$. There is no closed term $a\colon A$; otherwise, the application $c(a)$ would inhabit the empty type. We can use any known generalizations of Tait's computability method for system T to MLTT to re-prove the normalization theorem. For this, we need to extend the proof with an inductive clause that if a closed term $a\colon A$ is computable then so is $c(a)$, which is clearly true as there is no closed term of type A.

7.3 Models of dependent types in intensional MLTT

As mentioned earlier, we implemented several models of dependent types, in the sense of Dybjer's category with families (CwF) [29], for different purposes. It is, in fact, too early for us to claim that these are the internalized models of MLTT, because:

1. It is still a long-term goal to internalize the syntax of type theory in type theory [15].
2. It is a long standing conjecture that the syntactic category of MLTT is the initial object among CwFs and the others [75].

We do not have the syntax or the interpretation functors into the "models". Instead, for each model (or category more precisely), we defined (the interpretations of) contexts, substitutions, types and terms, and then verify the corresponding required rules/equations. This section presents the work using informal type theory. The formal formulation in Agda notation is available at [76].

7.3.1 The CwF of types

We studied the CwF-structure on the category of sets in Chapter 6.3. Here we formalize CwF-the structure on types. The formalization is very straightforward. In particular, (funext) is not needed to formalize the structure or to prove the equations. And all required equations hold judgmentally (in the sense that all the equality proofs are refl). Hence we made use of this to investigate the CwF-structure of identity types.

In the CwF of types, a context is a type and a substitution is a function of contexts. We write Cxt to denote the type of contexts and $\mathrm{Sub}(\Delta, \Gamma)$ to denote the type of substitutions of contexts Δ and Γ, and then have the following definitions:

$$\mathrm{Cxt} :\equiv \mathcal{U} \qquad \mathrm{Sub}(\Delta, \Gamma) :\equiv \Delta \to \Gamma.$$

A type in context Γ is a Γ-indexed type family, *i.e.*

$$\mathrm{Type}(\Gamma) :\equiv \Gamma \to \mathcal{U}.$$

A term of type A in context Γ is a dependent function mapping each $\gamma\colon \Gamma$ to an element of A_γ, *i.e.*

$$\mathrm{Term}(\Gamma, A) :\equiv \Pi(\gamma{:}\Gamma).A_\gamma.$$

Substitutions of types and of terms are defined using composition: given $A\colon \mathrm{Type}(\Gamma)$, $u\colon \mathrm{Term}(\Gamma, A)$ and $\sigma\colon \mathrm{Sub}(\Delta, \Gamma)$,

$$A[\sigma] :\equiv A \circ \sigma, \qquad u[\sigma] :\equiv u \circ \sigma.$$

Given $\Gamma\colon \mathrm{Cxt}$ and $A\colon \mathrm{Type}(\Gamma)$, the extended context $\Gamma.A$ is the dependent sum, *i.e.*

$$\Gamma.A :\equiv \Sigma(\gamma{:}\Gamma).A_\gamma.$$

The substitution $\mathrm{p}\colon \Gamma.A \to \Gamma$ and the term $\mathrm{q}\colon \mathrm{Term}(\Gamma.A, A[\mathrm{p}])$ are the first and second projections.

We omit the definitions of (the interpretations of) Π-type and of Σ-type, and probe into the one of identity type. Given $A\colon \mathrm{Type}(\Gamma)$, we know $\mathrm{Id}_A\colon \mathrm{Type}(\Gamma.A.A[\mathrm{p}])$ has to

be a type family indexed by tuples (γ, a, b) where $\gamma\colon \Gamma$ and $a, b\colon A_\gamma$, and hence define $\mathrm{Id}_A(\gamma, a, b)\colon \mathcal{U}$ to be the identity type of a and b, *i.e.*

$$\mathrm{Id}_A(\gamma, a, b) :\equiv a = b.$$

The reflexivity substitution $\mathrm{R}\colon \Gamma.A \to \Gamma.A.A[p].\mathrm{Id}_A$ maps an element to the reflexivity in the following sense:

$$\mathrm{R}(\gamma, a) :\equiv (\gamma, a, a, \mathrm{refl}).$$

Given a type $B\colon \mathrm{Type}(\Gamma.A.A[\mathrm{p}].\mathrm{Id}_A)$ and a term $u\colon \mathrm{Term}(\Gamma.A, B[R])$, we define the term $\mathrm{J}(u)\colon \mathrm{Term}(\Gamma.A.A[\mathrm{p}].\mathrm{Id}_A, B)$ using the J-eliminator as follows

$$\mathrm{J}(u)(\gamma, a, b, p) :\equiv \mathrm{J}(\lambda x.u(\gamma, x), a, b, p)$$

where the second J is the J-eliminator in the meta-theory discussed in Chapter 6.1. Moreover, all the equations we proposed for identity types in Chapter 6.3 type check and hold judgmentally.

7.3.2 The CwF of presheaves

For simplicity, here we discuss the formulation of presheaves on monoids only. The one of presheaves on arbitrary categories should be highly similar. The purpose of such formulation is to explore the feasibility of type-theoretic implementation of sheaf models. We also implemented the essential CwF-structure of sheaves on our uniform-continuity site (explained in Chapter 6.5) in Agda, which is available at [76].

A *naive* formulation of the type of presheaves on a monoid $(M, \circ, 1)$ is the following

$$\begin{aligned}\mathrm{Presheaf} :\equiv\ & \Sigma(\Gamma\colon \mathcal{U}).\\ & \Sigma(\cdot\colon \Gamma \to M \to \Gamma).\\ & \quad (\Pi(\gamma{:}\Gamma).\ \gamma \cdot 1 = \gamma)\\ & \quad \times(\Pi(\gamma{:}\Gamma).\Pi(t, r{:}M).\ (\gamma \cdot t) \cdot r = \gamma \cdot (t \circ r)).\end{aligned}$$

There is a problem in the above formulation. But we keep working on this naive formulation until the problem is identified. Natural transformations are formulated as

$$\mathrm{Nat}(\Delta, \Gamma) :\equiv \Sigma(\sigma\colon |\Delta| \to |\Gamma|).\ \Pi(\delta{:}|\Delta|).\Pi(t{:}M).\ \sigma(\delta) \cdot_\Gamma t = \sigma(\delta \cdot_\Delta t)$$

where $|\Gamma|\colon \mathcal{U}$ is the underlying type of $\Gamma\colon$ Presheaf, and $\cdot_\Gamma\colon |\Gamma| \to M \to |\Gamma|$ is the monoid action of Γ, both of which can be easily defined using projections. We omit the subscript and write $\cdot$ to denote the monoid action of any presheaf, by abuses of notations. Using projections again, we get the proofs of presheaf conditions of $\Gamma\colon$ Presheaf, and write Γ_1 and Γ_2 to denote them, *i.e.*

$$\Gamma_1\colon \Pi(\gamma{:}|\Gamma|).\ \gamma \cdot 1 = \gamma \qquad \Gamma_2\colon \Pi(\gamma{:}|\Gamma|).\Pi(t, r{:}M).\ (\gamma \cdot t) \cdot r = \gamma \cdot (t \circ r).$$

Contexts are (interpreted as) presheaves and substitutions are natural transformations:

$$\mathrm{Cxt} :\equiv \mathrm{Presheaf} \qquad \mathrm{Sub}(\Delta, \Gamma) :\equiv \mathrm{Nat}(\Delta, \Gamma).$$

Then types in context Γ are formulated as follows:

$$\begin{aligned}\mathrm{Type}(\Gamma) \quad :\equiv \quad & \Sigma(A\colon |\Gamma| \to \mathcal{U}).\\ & \Sigma(*\colon \Pi(\gamma\colon|\Gamma|).A_\gamma \to \Pi(t\colon M).A_{\gamma\cdot t}).\\ & \quad (\Pi(\gamma\colon|\Gamma|).\Pi(a\colon A_\gamma).\ a * 1 =^{\Gamma_1(\gamma)} a)\\ & \quad \times(\Pi(\gamma\colon|\Gamma|).\Pi(a\colon A_\gamma).\Pi(t,r\colon M).\ (a*t)*r =^{\Gamma_2(\gamma,t,r)} a*(t\circ r)).\end{aligned}$$

In the above formulation, we adopt the following abbreviations for the sake of readability:

- $a * t :\equiv\ *(\gamma, a, t)$.
  The first argument of the restriction map $*$ is omitted.
- $a * 1 =^{\Gamma_1(\gamma)} a\ :\equiv\ \mathrm{transport}^A(a*1, \Gamma_1(\gamma)) = a$.
  Since $a*1\colon A_{\gamma\cdot 1}$ and $a\colon A_\gamma$ are not in the same type, we have to transport $a*1$ along the path $\Gamma_1(\gamma)\colon \gamma\cdot 1 = \gamma$ in order to express the equality of $a*1$ and a.

For a given $A\colon \mathrm{Type}(\Gamma)$, we also write $|A|\colon |\Gamma| \to \mathcal{U}$ to denote its underlying type family, $*_A\colon \Pi(\gamma\colon|\Gamma|).|A|_\gamma \to \Pi(t\colon M).|A|_{\gamma\cdot t}$ to denote its restriction map, and A_1, A_2 to denote its proofs of presheaf conditions.

Type substitutions are defined using composition as in the CwF of types: Given $A\colon \mathrm{Type}(\Gamma)$ and $\sigma\colon \mathrm{Sub}(\Delta,\Gamma)$, we define the underlying type of $A[\sigma]$ by

$$|A[\sigma]|_\delta :\equiv |A|_{\mathrm{pr}_1(\sigma)(\delta)}$$

for $\delta \in |\Delta|$, and then restriction map $*_{A[\sigma]}\colon \Pi(\delta\colon|\Delta|).|A|_{\mathrm{pr}_1(\sigma)(\delta)} \to \Pi(t\colon M).|A|_{\mathrm{pr}_1(\sigma)(\delta\cdot t)}$ by

$$a *_{A[\sigma]} t :\equiv \mathrm{transport}^{|A|}(a *_A t, \mathrm{pr}_2(\sigma)(\delta, t)).$$

Since $a *_A t\colon |A|_{\mathrm{pr}_1(\sigma)(\delta)\cdot t}$, we have to transport it along the path $\mathrm{pr}_1(\sigma)(\delta)\cdot t = \mathrm{pr}_1(\sigma)(\delta\cdot t)$ given by the naturality (*i.e.* the second component) of σ. Then it remains to verify the presheaf conditions, in which the problem we mentioned above arises: To attempt to prove, for example, the first condition, *i.e.*

$$\Pi(\delta\colon|\Delta|).\Pi(a\colon|A|_{\mathrm{pr}_1(\sigma)(\delta)}).\ a *_{A[\sigma]} 1 =^{\Delta_1(\delta)} a,$$

we take any $\delta\colon|\Delta|$ and $a\colon|A|_{\mathrm{pr}_1(\sigma)(\delta)}$, and have

$$\begin{aligned} & \mathrm{transport}^{|A[\sigma]|}(a *_{A[\sigma]} 1, \Delta_1(\delta))\\ = \ & \mathrm{transport}^{|A[\sigma]|}(\mathrm{transport}^{|A|}(a *_A 1, \mathrm{pr}_2(\sigma)(\delta,1)), \Delta_1(\delta))\\ = \ & \mathrm{transport}^{|A|}(\mathrm{transport}^{|A|}(a *_A 1, \mathrm{pr}_2(\sigma)(\delta,1)), \mathrm{ap}_{\mathrm{pr}_1(\sigma)}(\Delta_1(\delta)))\\ = \ & \mathrm{transport}^{|A|}(a *_A 1, \mathrm{pr}_2(\sigma)(\delta,1)\cdot \mathrm{ap}_{\mathrm{pr}_1(\sigma)}(\Delta_1(\delta)))\end{aligned}$$

and

$$\mathrm{transport}^{|A|}(a *_A 1, \Gamma_1(\mathrm{pr}_1(\sigma)(\delta))) = a.$$

However, we cannot conclude $a *_{A[\sigma]} 1 =^{\Delta_1(\delta)} a$ unless we have

$$\mathrm{pr}_2(\sigma)(\delta,1)\cdot \mathrm{ap}_{\mathrm{pr}_1(\sigma)}(\Delta_1(\delta)) = \Gamma_1(\mathrm{pr}_1(\sigma)(\delta)).$$

There are three independent paths involved above, given by the naturality of σ and by the proofs of the first presheaf condition of Δ and Γ. To require two arbitrary paths to

be equal to another arbitrary one for a type X, whenever this type-checks, is equivalent to saying that X is a set (or an hset) in the sense that $\Pi(x, y{:}X).\ \Pi(p, q{:}\ x = y).\ p = q$.

Therefore, the underlying type of a presheaf has to be a set, and hence the formulation of presheaf become

$$\begin{aligned}\mathrm{Presheaf} :\equiv\ & \Sigma(\Gamma{:}\,\mathcal{U}).\\ & \Sigma(\cdot{:}\ \Gamma \to M \to \Gamma).\\ & \quad (\Pi(\gamma, \gamma'{:}\ \Gamma).\Pi(p, q{:}\ \gamma = \gamma').\ p = q)\\ & \times(\Pi(\gamma{:}\Gamma).\ \gamma \cdot 1 = \gamma)\\ & \times(\Pi(\gamma{:}\Gamma).\Pi(t, r{:}M).\ (\gamma \cdot t) \cdot r = \gamma \cdot (t \circ r)).\end{aligned}$$

Moreover, since a type is a family of presheaves, the underlying type family of a type has to be set-valued, and hence the proper formulation of type should be

$$\begin{aligned}\mathrm{Type}(\Gamma) :\equiv\ & \Sigma(A{:}\ |\Gamma| \to \mathcal{U}).\\ & \Sigma(*{:}\ \Pi(\gamma{:}|\Gamma|).A_\gamma \to \Pi(t{:}M).A_{\gamma\cdot t}).\\ & \quad (\Pi(\gamma{:}|\Gamma|).\Pi(a, a'{:}\ A_\gamma).\Pi(p, q{:}\ a = a').\ p = q)\\ & \times(\Pi(\gamma{:}|\Gamma|).\Pi(a{:}A_\gamma).\ a * 1 =^{\Gamma_1(\gamma)} a)\\ & \times(\Pi(\gamma{:}|\Gamma|).\Pi(a{:}A_\gamma).\Pi(t, r{:}M).\ (a * t) * r =^{\Gamma_2(\gamma,t,r)} a * (t \circ r)).\end{aligned}$$

The remaining constructions of the CwF-structure on presheaves are the straightforward translations of those in Chapter 6.5 (ignoring the sheaf condition). However, the proofs (*e.g.* the verifications of the presheaf conditions) are not easy, as they highly use the transport function and a number of its properties. We also remark that (funext) is only needed in the construction of Π-type, *e.g.* to prove that the proposed underlying type family of a Π-type is set-valued.

7.3.3 The CwF of C-spaces

One formulation of C-spaces has been given in Section 7.2.1. In this section, we take another formulation of the sheaf condition which is equivalent to the one adopted in Section 7.2.1 when (funext) is available. Since this is an exploration of internalizing models of type theory, we do not worry about the issues of (funext) too much and hence choose the following formulation of C-spaces that is more convenient to work with:

$$\begin{aligned}\mathrm{Space} :\equiv\ & \Sigma(X{:}\,\mathcal{U}).\\ & \Sigma(P{:}\ (\mathbf{2}^{\mathbb{N}} \to X) \to \mathcal{U}).\\ & \quad (\Pi(x{:}X).\ \lambda\alpha.x \in P)\\ & \times(\Pi(p{:}\ \mathbf{2}^{\mathbb{N}} \to X).\ p \in P \to \Pi(t{:}\ \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}).\ t \in \mathrm{C} \to p \circ t \in P)\\ & \times(\Pi(\rho{:}\ \mathbf{2} \to \mathbf{2}^{\mathbb{N}} \to X).(\Pi(i{:}\mathbf{2}).\rho_i \in P) \to \mathrm{Aml}(\rho) \in P)\end{aligned}$$

where $\mathrm{Aml}(\rho)$ is the amalgamation of ρ_0 and ρ_1 which is defined by

$$\mathrm{Aml}(\rho)(\alpha) :\equiv \rho_{\alpha_0}(\alpha \circ \mathrm{succ}).$$

And the type of continuous maps of X, Y: Space is formulated as

$$\mathrm{Map}(X, Y) :\equiv \Sigma(f{:}\ |X| \to |Y|).\ \Pi(p{:}\ \mathbf{2}^{\mathbb{N}} \to |X|).\ p \in \mathrm{Probe}(X) \to f \circ p \in \mathrm{Probe}(Y).$$

Again we use $|X|$ to represent the underlying type of $X : \mathrm{Space}$ and $\mathrm{Probe}(X)$ to represent its C-topology. We also write X_1, X_2, X_3 to denote the proofs of the three probe axioms of X.

As discussed in Chapter 6.4, contexts are interpreted as C-spaces and substitutions as continuous maps, *i.e.*

$$\mathrm{Cxt} :\equiv \mathrm{Space}, \qquad \mathrm{Sub}(\Delta, \Gamma) :\equiv \mathrm{Map}(\Delta, \Gamma).$$

Intuitively, a type in $\Gamma : \mathrm{Cxt}$ is a Γ-indexed family of "spaces": the underlying type family is indexed by elements of Γ and the C-topology family is indexed by the probes on Γ. Let the underlying type family of a type be $A : |\Gamma| \to \mathcal{U}$. A C-topology family Q on A should be indexed by maps $p : \mathbf{2}^{\mathbb{N}} \to |\Gamma|$ and probe witnesses $p \in \mathrm{Probe}(\Gamma)$, *i.e.*

$$Q : \Pi(p : \mathbf{2}^{\mathbb{N}} \to |\Gamma|).\ p \in \mathrm{Probe}(\Gamma) \to (\Pi(\alpha : \mathbf{2}^{\mathbb{N}}).A_{p(\alpha)}) \to \mathcal{U}.$$

Chapter 6.4 defines the three dependent probe axioms of the C-topology family Q on a type A in context Γ to be:

1. For all $\gamma \in \Gamma$ and $a \in A_\gamma$, the map $\lambda\alpha.a$ is in $Q_{\lambda\alpha.\gamma}$.
2. If $p \in \mathrm{Probe}(\Gamma)$, $q \in Q_p$ and $t \in \mathrm{C}$, then $q \circ t \in Q_{p \circ t}$.
3. For any $p_0, p_1 \in \mathrm{Probe}(\Gamma)$, $q_0 \in Q_{p_0}$ and $q_1 \in Q_{p_1}$, we have $q \in Q_p$ where p is defined by $p(i\alpha) = p_i(\alpha)$ and q is defined by $q(i\alpha) = q_i(\alpha)$.

Under the Curry-Howard interpretation, they become the following product type:

$$\begin{aligned}
&(\Pi(\gamma : |\Gamma|).\ \Pi(a : A_\gamma).\ \lambda\alpha.a \in Q_{(\lambda\alpha.\gamma, \Gamma_1(\gamma))}) \\
\times\ &(\Pi(p : \mathbf{2}^{\mathbb{N}} \to |\Gamma|).\ \Pi(p\Gamma : p \in \mathrm{Probe}(\Gamma)).\ \Pi(t : \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}).\ \Pi(tC : t \in \mathrm{C}). \\
&\ \Pi(q : \Pi(\alpha : \mathbf{2}^{\mathbb{N}}).A_{p(\alpha)}).q \in Q_{(p,p\Gamma)} \to q \circ t \in Q_{(p \circ t, \Gamma_2(p,p\Gamma,t,tC))}) \\
\times\ &(\Pi(\rho : \mathbf{2} \to \mathbf{2}^{\mathbb{N}} \to |\Gamma|).\ \Pi(\rho\Gamma : \Pi(i : \mathbf{2}).\rho_i \in \mathrm{Probe}(\Gamma)). \\
&\ \Pi(\zeta : \Pi(i : \mathbf{2}).\Pi(\alpha : \mathbf{2}^{\mathbb{N}}).A_{\rho_i(\alpha)}).(\Pi(i : \mathbf{2}).\zeta_i \in Q_{(\rho_i, \rho\Gamma_i)}) \to \mathrm{Aml}(\zeta) \in Q_{(\mathrm{Aml}(\rho), \Gamma_3(\rho,\rho\Gamma))}).
\end{aligned}$$

We write it as

$$\text{probe-axioms}^\Gamma(A, Q)$$

and formulate types in context Γ as

$$\begin{aligned}
\mathrm{Type}(\Gamma) \quad :\equiv \quad & \Sigma(A : |\Gamma| \to \mathcal{U}). \\
& \Sigma(Q : \Pi(p : \mathbf{2}^{\mathbb{N}} \to |\Gamma|).\ p \in \mathrm{Probe}(\Gamma) \to (\Pi(\alpha : \mathbf{2}^{\mathbb{N}}).A_{p(\alpha)}) \to \mathcal{U}). \\
& \ \text{probe-axioms}^\Gamma(A, Q).
\end{aligned}$$

And terms of type A in context Γ are formulated as

$$\begin{aligned}
\mathrm{Term}(\Gamma, A) \quad :\equiv \quad & \Sigma(u : \Pi(\gamma : |\Gamma|).|A|_\gamma). \\
& \Pi(p : \mathbf{2}^{\mathbb{N}} \to |\Gamma|).\ \Pi(p\Gamma : p \in \mathrm{Probe}(\Gamma)).\ u \circ p \in \mathrm{Probe}(A)_{(p,p\Gamma)}
\end{aligned}$$

where similarly $|A|$ denotes the underlying type family of type A and $\mathrm{Probe}(A)$ denotes its C-topology family.

However, there is a problem in the above formulations: the C-topology family of a type is also indexed by probe witnesses, but there may be more than one probe witnesses for the same map. For example, given $p\colon \mathbf{2}^{\mathbb{N}} \to |\Gamma|$ and $p\Gamma\colon p \in \mathrm{Probe}(\Gamma)$ for some $\Gamma\colon \mathrm{Cxt}$, we have another probe witness $p\Gamma' :\equiv \Gamma_2(p, p\Gamma, 1, 1\mathrm{C})$, where 1C is the uniform-continuity witness of the identity map $1\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$. If Q is the C-topology family of $A\colon \mathrm{Type}(\Gamma)$, then it does not make sense to have $Q_{(p,p\Gamma)}$ and $Q_{(p.p\Gamma')}$ two distinct C-topologies, because Q should be indexed exactly by (the underlying maps of) the probes rather than the proofs that they are probes. This is more than a conceptual issue. As explained in Chapter 6.4, type substitutions are defined by compositions: given $A\colon \mathrm{Type}(\Gamma)$ and $\sigma\colon \mathrm{Sub}(\Delta, \Gamma)$, the underlying type family of $A[\sigma]\colon \mathrm{Type}(\Delta)$ is defined by

$$|A[\sigma]|_\delta :\equiv |A|_{\mathrm{pr}_1(\sigma)(\delta)}$$

for $\delta\colon |\Delta|$, and the C-topology family is defined by

$$q \in \mathrm{Probe}(A[\sigma])_{(p,p\Delta)} :\equiv q \in \mathrm{Probe}(A)_{(\mathrm{pr}_1(\sigma)\circ p, \mathrm{pr}_2(\sigma)(p,p\Delta))}$$

for $p\colon \mathbf{2}^{\mathbb{N}} \to |\Delta|$, $p\Delta\colon p \in \mathrm{Probe}(\Delta)$ and $q\colon \Pi(\alpha\colon \mathbf{2}^{\mathbb{N}}).|A|_{\mathrm{pr}_1(\sigma)(p(\alpha))}$. When verifying the dependent probe axioms for $A[\sigma]$, we got into a situation where we had $q \in \mathrm{Probe}(A)_{(p,p\Gamma)}$ which is not sufficient to conclude $q \in \mathrm{Probe}(A)_{(p,p\Gamma')}$ since $p\Gamma, p\Gamma'\colon p \in \mathrm{Probe}(\Gamma)$ are obtained via different ways (*e.g.* using the naturality of σ or the probe axioms of Γ).

One way to address the above issue is to additionally require C-topologies to be proposition-valued: If P is a C-topology of a space Γ then

$$\Pi(p\colon \mathbf{2}^{\mathbb{N}} \to \Gamma).\ \mathrm{isProp}(p \in P)$$

where $\mathrm{isProp}(X) :\equiv \Pi(x, x'\colon X).\ x = x'$ expresses that $X\colon \mathcal{U}$ is a proposition. Similarly, if Q is a C-topology family of a type A in context Γ then

$$\Pi(p\colon \mathbf{2}^{\mathbb{N}} \to |\Gamma|).\ \Pi(p\Gamma\colon p \in \mathrm{Probe}(\Gamma)).\ \Pi(q\colon \Pi(\alpha\colon \mathbf{2}^{\mathbb{N}}).A_{p(\alpha)}).\ \mathrm{isProp}(q \in Q_{(p,p\Gamma)}).$$

Requiring C-topologies to be proposition-valued is reasonable when we think of a C-topology on X to be a *subset* of the function space $\mathbf{2}^{\mathbb{N}} \to X$. In type theory $\Sigma(x\colon X).P(x)$ is called a subtype of $X\colon \mathcal{U}$, for some proposition-valued type family $P\colon X \to \mathcal{U}$. Here P has to be proposition-valued, otherwise the subtype may have more elements than X.

In our Agda implementation, we instead added the following condition

$$\Pi(p\colon \mathbf{2}^{\mathbb{N}} \to |\Gamma|).\ \Pi(p\Gamma, p\Gamma'\colon p \in \mathrm{Probe}(\Gamma)).\Pi(q\colon \Pi(\alpha\colon \mathbf{2}^{\mathbb{N}}).A_{p(\alpha)}).\ q \in Q_{(p,p\Gamma)} \to q \in Q_{(p,p\Gamma')}$$

in the product type $\text{probe-axioms}^\Gamma(A, Q)$. If C-topologies are proposition-valued, then clearly this condition always holds. But this weaker condition concisely solves the above issue, and is preserved by all the constructions of the CwF-structure on C-spaces, *e.g.* type substitution, Σ-type and Π-type.

CHAPTER 8

Construction of the model in Agda

Agda [11, 12, 63] is both a *functional programming language* with dependent types, based on Martin-Löf type theory (MLTT), and a *proof assistant* for writing and checking proofs, based on the Curry–Howard interpretation of logic. We confine ourselves to a fragment of Agda corresponding to intensional MLTT, with an extension to address the issues caused by the absence of function extensionality and by the presence of proof relevance of MLTT as discussed in Section 7.1. In particular, we disable Streicher's K axiom [69, 44, 20, 18], which is enabled by default in Agda.

Our Agda implementation, which is available at [76], should be considered as a significant part of this thesis. Not only did it take a considerable portion of the research time, but also it demonstrates how we achieved the main aim of the thesis, that is to extract computational content of type-theoretic proofs which use the uniform-continuity principle.

We begin with a short introduction to Agda (Section 8.1), with the necessary definitions to support the discussion in this chapter. Then we list some examples in our Agda implementation (Section 8.2), regarding the constructions of C-spaces and of system T. Using the example code, we demonstrate how to "run" our model to compute uniform-continuity moduli of T-definable functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ (Section 8.3). In the end of the chapter, we conclude with an overview of our Agda files (Section 8.4).

8.1 Brief introduction to Agda

A data type in Agda can be defined by using the keyword `data` to specify its name and using `where` to list its constructors. For instance, nature numbers are defined as follows:

```
data ℕ : Set where
 zero : ℕ
 succ : ℕ → ℕ
```

Agda allows Unicode characters in naming identifiers, as shown in the above example. The important point is that the *type* of the data type that we are introducing must be stated explicitly. Here `Set` is the *type* of *small types*. Agda has countably many type universes

$$\texttt{Set} : \texttt{Set}_1 : \texttt{Set}_2 : \ldots$$

For example, $\mathbb{N}$ is an element of `Set` as defined above; while C-spaces live in $\texttt{Set}_1$ as we will see in Section 8.2.

One example of a dependent type is that of vectors:

```
data Vec (A : Set) : ℕ → Set where
 ⟨⟩ : Vec A zero
 _::_ : {n : ℕ} → A → Vec A n → Vec A (succ n)
```

where the first constructor $\langle\rangle$ represents an empty vector, and the second one `_::_` constructs a vector over type `A` of length `succ n` from a given element of `A` and a given vector over `A` of length `n`. In the above definition, `A` is a *parameter* to `Vec`, which means it is fixed over all constructors of `Vec`. The type `Vec` also has an *index* of type $\mathbb{N}$ which denotes the length of the vector and can vary over constructors. Agda allows *implicit arguments*, declared by enclosing them in curly bracket, and tries to infer them automatically. For instance, the first argument `n` of the constructor `_::_` is implicit. Underscores in Agda are used to denote the positions of arguments, which allows to define prefix, postfix and even midfix operators. For instance "`x :: xs`" is the same as "`_::_ x xs`".

The type former of dependent products (or Π-types) is already pre-defined. We can also redefine it as follows:

```
Π : {A : Set} → (A → Set) → Set
Π {A} B = (a : A) → B a
```

Dependent sums (or Σ-types) can be defined as follows:

```
data Σ {A : Set} (B : A → Set) : Set where
 _,_ : (a : A) (b : B a) → Σ B
```

In both definitions, we make the argument `A` implicit. Then we write, for example, `Σ \(a : A) → B a` instead of `Σ B` to have a notation closer to $\Sigma(a{:}A).B(a)$ that is used in Chapter 7. Projections are defined by pattern matching:

```
pr₁ : {A : Set} {B : A → Set} → (Σ \(a : A) → B a) → A
pr₁ (a , _) = a

pr₂ : {A : Set} {B : A → Set} → (w : Σ \(a : A) → B a) → B(pr₁ w)
pr₂ (_ , b) = b
```

where the underscores denote nameless arguments that are not used. Recall that product types can be defined as Σ-types:

```
_×_ : Set → Set → Set
A × B = Σ \(a : A) → B
```

Coproducts (or disjoint unions) can be defined as follows, where `case` is the non-dependent elimination rule:

```
data _+_ (A B : Set) : Set where
 inl : A → A + B
 inr : B → A + B

case : {X₀ X₁ Y : Set} → (X₀ → Y) → (X₁ → Y) → X₀ + X₁ → Y
case f₀ f₁ (inl x₀) = f₀ x₀
case f₀ f₁ (inr x₁) = f₁ x₁
```

The empty type is defined without given constructors:

```
data ∅ : Set where

∅-elim : {A : Set} → ∅ → A
∅-elim ()
```

Here is a special pattern `()` that indicates that there is no case to be matched. Identity types and the J-eliminator are defined as follows:

```
data _≡_ {A : Set} : A → A → Set where
 refl : {a : A} → a ≡ a

J : {A : Set} (C : (x y : A) → x ≡ y → Set)
  → ((a : A) → C a a refl) → (x y : A) (p : x ≡ y) → C x y p
J C c x .x refl = c x
```

The use of a dot is a technicality of the implementation of Agda to deal with non-linear patterns. It is needed because the only constructor of the identity type is `refl`, which then forces `x` and `y` to be the same (definitionally equal).

Agda provides a form of dependent pattern matching [20], which allows one to prove Streicher's K axiom [44], and hence that every type in Agda is a set (the equality of any two of its points is a proposition). A version of pattern matching without the K axiom [18] is enabled with the flag `--without-K`, by adding the following to the Agda code:

```
{-# OPTIONS --without-K #-}
```

As mentioned earlier, we have chosen to implement the model without using the K axiom, which is ensured by using this without-K flag in every Agda file, in order to be more general and also to be compatible with homotopy type theory [72].

8.2 Excerpts of the Agda implementation

In this section, we present some parts of our Agda implementation that correspond to the simplest approach of postulating (funext) explained in Chapter 7.2.1 and can be found in the package `UsingFunext` at [76]. One purpose of presenting the Agda implementation is to compare our Agda programs to the informal proofs in Chapter 7. Moreover, the Agda code presented in this section will also support the demonstration of running our model to compute uniform-continuity moduli in Section 8.3.

The notations used in this section are highly close to those in Chapter 7. For example, we also use `C` to denote the type of uniform continuity of maps $\mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ and write `t ∈ C` to express that `t` is uniformly continuous as in Chapter 7. Additional explanations will be provided when necessary.

C-spaces and continuous maps are formulated as follows:

```
probe-axioms : (X : Set) → ((₂ℕ → X) → Set) → Set
probe-axioms X P =
   ((x : X) → (λ α → x) ∈ P)
 × ((t : ₂ℕ → ₂ℕ) → t ∈ C → (p : ₂ℕ → X) → p ∈ P → p ∘ t ∈ P)
 × ((p : ₂ℕ → X) → (Σ \(n : ℕ) → (s : ₂Fin n) → p ∘ cons s ∈ P) → p ∈ P)
```

```
TopologyOn : Set → Set₁
TopologyOn X = Σ \(P : (₂ℕ → X) → Set) → probe-axioms X P

Space : Set₁
Space = Σ \(X : Set) → TopologyOn X

continuous : (X Y : Space) → (U X → U Y) → Set
continuous X Y f = (p : ₂ℕ → U X) → p ∈ Probe X → f ∘ p ∈ Probe Y

Map : Space → Space → Set
Map X Y = Σ \(f : U X → U Y) → continuous X Y f
```

where

```
U     : Space → Set
Probe : (X : Space) → (₂ℕ → U X) → Set
```

are defined using the projections $\mathtt{pr}_1$ and $\mathtt{pr}_2$ to extract the underlying set and the C-topology of a `Space`.

Remark. In Chapter 7.2, we write continuous(f) to mean that f is continuous, using informal type theory. Here we have to specify the domain and codomain explicitly and write `continuous X Y f` because, though Agda allows users to use implicit arguments, we found that in practice it cannot infer spaces when making the first two arguments of `continuous` implicit. In a few other places of the development, we have to specify some arguments explicitly; therefore, the Agda implementation is in fact longer and less readable than the informal development in Chapter 7.2.

The fan functional is a continuous map of the following type:

```
fan : Map ((ℕSpace ⇒ ₂Space) ⇒ ℕSpace) ℕSpace
```

where `ℕSpace` and `₂Space` are the discrete spaces of natural numbers and of booleans, and `X ⇒ Y` is the exponential of `X Y : Space`. The underlying map (*i.e.* the first component) of `fan` is defined as follows:

```
|fan| : Map (ℕSpace ⇒ ₂Space) ℕSpace → ℕ
|fan| φ = pr₁ (pr₂ (Lemma[Yoneda] ℕSpace φ))
```

where `Lemma[Yoneda]` is (one direction of) the Yoneda Lemma that each continuous map $\mathbf{2}^{\mathbb{N}} \to X$ corresponds to a probe on X

```
Lemma[Yoneda] : ∀(X : Space) → Map (ℕSpace ⇒ ₂Space) X
              → Σ \(p : ₂ℕ → U X) → p ∈ Probe X
```

and is proved by compositing the given continuous map to the "identity" probe on the exponential $\mathbf{2}^{\mathbb{N}}$ (also see Chapter 7.2.1). We remark that the construction of the continuity witness of `|fan|`, that is the second component of `fan`, unavoidably requires (funext) or its weakening ¬¬(funext) as explained in Chapters 7.1 and 7.2.4.

The term language of System T is implemented with three components: types, contexts and terms. Here we consider ground types ② of booleans and Ⓝ of natural numbers, and function types σ ⇛ τ, which are formalized by the following data type:

```
data Ty : Set where
 ② : Ty
 Ⓝ : Ty
 _⇛_ : Ty → Ty → Ty
```

To avoid naming variables, we use de Bruijn indices to represent variables in T-terms, by defining contexts as vectors with a projection function to give the types of variables in a given context:

```
data Cxt : ℕ → Set where
 ε : Cxt zero
 _+_ : {n : ℕ} → Cxt n → Ty → Cxt (succ n)

_[_] : {n : ℕ} → Cxt n → Fin n → Ty
(xs + x) [ zero ]   = x
(xs + x) [ succ i ] = xs [ i ]
```

Here `Fin n` is a finite type of size `n`. Then we define terms by

```
data Tm : {n : ℕ} → Cxt n → Ty → Set where
 VAR  : {n : ℕ}{Γ : Cxt n}           → (i : Fin n) → Tm Γ (Γ [ i ])
 ⊥    : {n : ℕ}{Γ : Cxt n}           → Tm Γ ②
 ⊤    : {n : ℕ}{Γ : Cxt n}           → Tm Γ ②
 IF   : {n : ℕ}{Γ : Cxt n}{σ : Ty}   → Tm Γ (σ ⇛ σ ⇛ ② ⇛ σ)
 ZERO : {n : ℕ}{Γ : Cxt n}           → Tm Γ Ⓝ
 SUCC : {n : ℕ}{Γ : Cxt n}           → Tm Γ (Ⓝ ⇛ Ⓝ)
 REC  : {n : ℕ}{Γ : Cxt n}{σ : Ty}   → Tm Γ (σ ⇛ (Ⓝ ⇛ σ ⇛ σ) ⇛ Ⓝ ⇛ σ)
 LAM  : {n : ℕ}{Γ : Cxt n}{σ τ : Ty} → Tm (Γ + σ) τ → Tm Γ (σ ⇛ τ)
 _•_  : {n : ℕ}{Γ : Cxt n}{σ τ : Ty} → Tm Γ (σ ⇛ τ) → Tm Γ σ → Tm Γ τ
```

In Chapter 5.1 we add a constant of the fan functional for the purpose of formulating the uniform-continuity principle. Here we also extend the above theory with the following `Tm`-constructor

```
FAN  : {n : ℕ} {Γ : Cxt n} → Tm Γ ((((Ⓝ ⇛ ②) ⇛ Ⓝ) ⇛ Ⓝ) ⇛ Ⓝ)
```

in order to have more interesting sample computations of uniform-continuity moduli of T-definable functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$, as demonstrated in Section 8.3.

The interpretation of system T in C-spaces is given via three interpretation maps which have the following types:

```
⟦_⟧ʸ : Ty → Space
⟦_⟧ᶜ : {n : ℕ} → Cxt n → Space
⟦_⟧ᵐ : {n : ℕ}{Γ : Cxt n}{σ : Ty} → Tm Γ σ → Map ⟦ Γ ⟧ᶜ ⟦ σ ⟧ʸ
```

In particular, the additional `Tm`-constructor is interpreted using the functional `fan`:

```
⟦ FAN ⟧ᵐ = (λ _ → fan) , (λ _ _ → λ p pP _ _ → pr₂ fan p pP)
```

In words, the `Tm`-constructor `FAN` is interpreted as a constant function with value `fan`, whose continuity witness is constructed using the one of `fan`.

8.3 Sample computations of least moduli of uniform continuity

Our type-theoretic developments of C-spaces and of system T including its interpretation in C-spaces allow us to compute least moduli of uniform continuity of functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ from their T-definitions, simply via the following:

```
modu : Tm ε (((Ⓝ ⇛ ②) ⇛ Ⓝ) → ℕ
modu F = pr₁ fan (pr₁ ⟦ F ⟧ᵐ ⋆)
```

where $\star$ is the unique element of the singleton type which interprets the empty context. In words, any T-term `F : Tm ε (((Ⓝ ⇛ ②) ⇛ Ⓝ)` is interpreted as a continuous map `pr₁ ⟦ F ⟧ᵐ ⋆ : Map (ℕSpace ⇛ ₂Space) ℕSpace`. Then applying the functional `fan` (defined and explained in Chapter 7.2.1 using informal type theory and in Section 8.2 using Agda notation) to the continuous map gives its least modulus of uniform continuity.

In this section we provide some examples of close T-terms of type `(Ⓝ ⇛ ②) ⇛ Ⓝ` and the expected results of applying `modu` to them. As discussed in Chapter 7, all our formalizations, except `UsingFunext`, give the expected results.

To improve the readability of our examples, we defined the following T-numerals

```
ONE TWO THREE FOUR FIVE : {n : ℕ}{Γ : Cxt n} → Tm Γ Ⓝ
ONE   = SUCC • ZERO
TWO   = SUCC • ONE
THREE = SUCC • TWO
FOUR  = SUCC • THREE
FIVE  = SUCC • FOUR
```

as well as the interpretation $[\![\_]\!]$ of closed T-terms with meaning in the function space $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$

```
⟦_⟧ : Tm ε (((Ⓝ ⇛ ②) ⇛ Ⓝ) → ₂ℕ → ℕ
⟦ t ⟧ α = pr₁ (pr₁ ⟦ t ⟧ᵐ ⋆) (ID α)
```

where `ID : ₂ℕ → U(ℕSpace ⇛ ₂Space)` is the "identity" probe of the exponential $\mathbf{2}^{\mathbb{N}}$, as discussed in Chapter 7.2.1, which sends a sequence to a continuous map. In each of the following examples, both a T-term and its type-theoretic meaning are provided:

1. $\mathtt{F}_1$ is a constant function which always returns the numeral `TWO`:

   ```
   F₁ = LAM TWO

   F₁-interpretation : ⟦ F₁ ⟧ ≡ λ α → 2
   F₁-interpretation = refl
   ```

   The expected result of normalizing `modu` $\mathtt{F}_1$ is `0`.

2. $\mathtt{F}_2$ is also constant, though it looks at the first bit of inputs:

   ```
   F₂ = LAM (IF • (VAR zero • ZERO) • ONE • ONE)

   F₂-interpretation : ⟦ F₂ ⟧ ≡ λ α → if (α 0) 1 1
   F₂-interpretation = refl
   ```

The expected result of normalizing `modu F₂` is also `0`.

3. `F₃` is a non-trivial function defined using the case-distinction function `IF`:

   It returns `FIVE`, if the 4th bit is $\bot$.
   It returns `ONE`, if the 4th bit is $\top$ and the 5th one is $\bot$.
   It returns `TWO`, if both the 4th and 5th bits are $\top$.

```
F₃ = LAM (IF • (VAR zero • THREE) • FIVE
             • (IF • (VAR zero • FOUR) • ONE • TWO))

F₃-interpretation : ⟦ F₃ ⟧ ≡ λ α → if (α 3) 5 (if (α 4) 1 2)
F₃-interpretation = refl
```

   The expected result of normalizing `modu F₃` is `5`.

4. `F₄` is another constant function:

   It looks at 2nd and 3rd or 4th bits but always returns `ZERO`.

```
F₄ = LAM (IF • (VAR zero • ONE) • (IF • (VAR zero • TWO) • ZERO • ZERO)
             • (IF • (VAR zero • THREE) • ZERO • ZERO))

F₄-interpretation : ⟦ F₄ ⟧ ≡ λ α → if (α 1) (if (α 2) 0 0)
                                        (if (α 3) 0 0)
F₄-interpretation = refl
```

   The expected result of normalizing `modu F₄` is `0`.

5. `F₅` is a non-trivial function defined using the recursor `REC`:

   It applies `SUCC` to `ZERO` 3 times, *i.e.* returns `THREE`, if the 2nd bit is $\bot$.
   It applies `SUCC` to `ZERO` twice, *i.e.* returns `TWO`, if the 2nd bit is $\top$.

```
F₅ = LAM (REC • ZERO • LAM SUCC • (IF • (VAR zero • ONE) • THREE • TWO))

F₅-interpretation : ⟦ F₅ ⟧ ≡ λ α → rec zero (λ _ → succ) (if (α 1) 3 2)
F₅-interpretation = refl
```

   The expected result of normalizing `modu F₅` is `2`.

6. `F₆` is a more complicated example defined using `F₄` and `F₅`:

```
F₆ = LAM (REC • (IF • (VAR zero • (F₅ • VAR zero)) • FIVE • TWO) • LAM SUCC
             • (IF • (VAR zero • (F₄ • VAR zero)) • THREE • TWO))

F₆-interpretation : ⟦ F₆ ⟧ ≡ λ α → rec (if (α (⟦ F₅ ⟧ α)) 5 2) (λ _ → succ)
                                       (if (α (⟦ F₄ ⟧ α)) 3 2)
F₆-interpretation = refl
```

   The expected result of normalizing `modu F₆` is `4`.

8.4 Overview of the Agda implementation

The Agda implementation is organized in the following packages:

1. `Preliminaries` is a general purpose library. We do not use the Agda standard library because it assumes the K axiom.

2. `Continuity` contains (1) the formal proof of Theorem 2.3.1 that the two type-theoretic formulations of the uniform-continuity principle are logically equivalent, and (2) some properties of uniform continuity for constructing the model such as the closure property under composition.

3. There are two sub-packages in `UsingFunext`:

   (1) `Space` implements the main constructions of C-spaces, including the (local) cartesian closedness, discrete C-spaces and the fan functional, with the aid of the postulated constant of function extensionality (funext). This can be regarded as a formalization of the informal development in Chapters 3.3–3.5 within intensional MLTT extended with (funext).

   (2) `ModellingUC` contains the formal proofs of

   - Theorem 4.3.5: the full type hierarchy is equivalent to the Kleene–Kreisel continuous hierarchy within C-spaces when assuming the principle of uniform continuity and function extensionality.
   - Theorem 5.1.3: all T-definable maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous;
   - Theorem 5.1.4: the model of C-spaces validates the uniform-continuity principle in system T;
   - Theorem 5.2.2: the uniform-continuity principle in HA^{ω} can be realized by the fan functional;
   - Theorem 6.2.1: the Curry–Howard formulation of the uniform-continuity principle is inhabited in the locally cartesian closed category of C-spaces.

   As explained in Chapter 7.2.1, the computational content of the implementation in this package is destroyed because the postulated constant (funext) has no computational rule. For example, in the module `ComputationExperiments`, the closed Agda term representing a uniform-continuity modulus of the constant map $\lambda\alpha.\mathrm{if}(\mathtt{t}, 2, 2)$ cannot be normalized to a numeral (its normal form has 367 lines in the way Agda outputs it).

4. The following packages also provide the constructions and proofs of C-spaces, and show how the uniform-continuity principle in system T is validated in the model of C-spaces, with different approaches to address the issues caused by the lack of function extensionality in MLTT (and in Agda):

   - `UsingSetoid` makes use of setoids, as discussed in Chapter 7.2.2, and hence does not postulate any form of function extensionality, but produces a long and unreadable formalization.

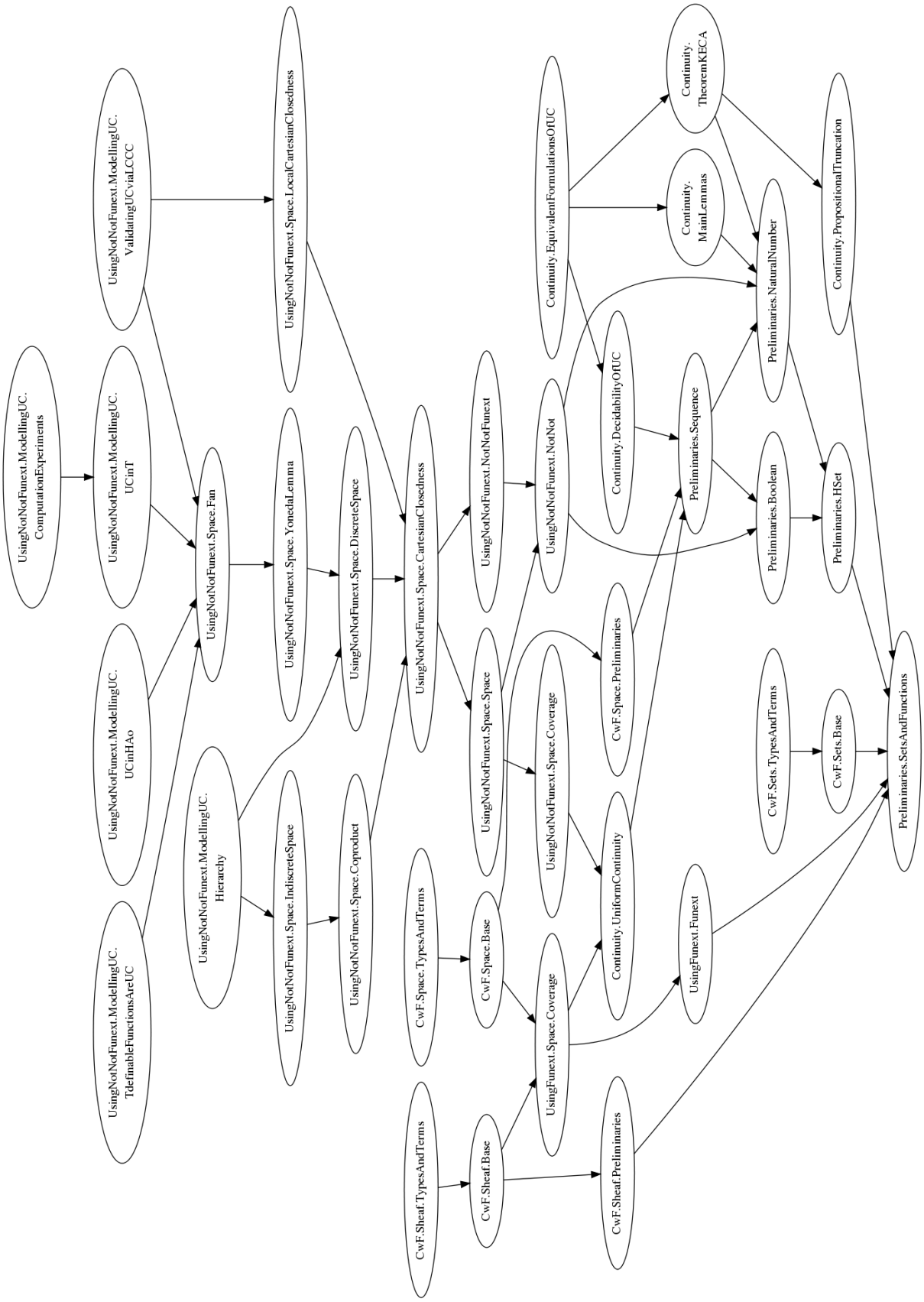
- `AddingProbeAxiom` introduces an additional probe axiom (that if a map is pointwise equal to a probe then it is a probe) in the definition of C-spaces as discussed in Chapter 7.2.3. We remark that this package uses the postulated constant of function extensionality.
- `UsingIrrelevantFunext` postulates function extensionality within Agda's irrelevant field. It also requires an additional probe axiom that is stronger than the one mentioned above, and hence produces a model that is not exactly the same as the one in `UsingFunext` a constructive point of view.
- `UsingNotNotFunext` postulates the double negation of function extensionality to construct the model, as discussed in Chapter 7.2.4. It produces the same interpretation to simple (and dependent) types and preserves the computational content of the development.

Each of the above packages has a module named *ComputationExperiments*, which contains some sample computations of minimal moduli of uniform continuity of T-definable functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ which are explained in Section 8.3. We emphasize that the uniform-continuity moduli (that is a closed Agda term) of the all sample functions in these modules can be successfully normalized to numerals.

5. `CwF` formalizes the structures of category with families on sets/types, on C-spaces, and on sheaves.

In total, the above Agda formalization has over 16,000 lines split into 85 modules. The following graph shows the dependency of the modules for the version `UsingNotNotFunext`:

Figure 8.1: A dependency graph of the modules for the version `UsingNotNotFunext`.



CHAPTER 9

Summary and further work

We conclude the thesis with a summary of the main results in the thesis and a few interesting directions for further research that are related to the thesis.

9.1 Continuity principles in type theory

In Chapter 2 we study the compatibility of intensional Martin-Löf type theory (MLTT) with the following two Brouwerian principles of continuity:

(Cont) All functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are continuous.

(UC) All functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous.

Their precise formulations

(Cont) $\forall(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}).\ \forall(\alpha{:}\mathbb{N}^{\mathbb{N}}).\ \exists(m{:}\mathbb{N}).\ \forall(\beta{:}\mathbb{N}^{\mathbb{N}}).\ \alpha =_m \beta \Rightarrow f\alpha = f\beta,$

(UC) $\forall(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \exists(m{:}\mathbb{N}).\ \forall(\alpha,\beta{:}\mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \Rightarrow f\alpha = f\beta.$

are validated by Johnstone's topological topos [48]. But what we are interested in is their Curry–Howard interpretations:

(CH-Cont) $\Pi(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}).\ \Pi(\alpha{:}\mathbb{N}^{\mathbb{N}}).\ \Sigma(m{:}\mathbb{N}).\ \Pi(\beta{:}\mathbb{N}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta,$

(CH-UC) $\Pi(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \Sigma(m{:}\mathbb{N}).\ \Pi(\alpha,\beta{:}\mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta.$

The principle (CH-Cont) is provably false in (intensional and hence in extensional) MLTT [31], since using (CH-Cont) one can define a modulus-of-continuity function, which allows one to define non-continuous functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$. An informal proof of this fact is recalled in Chapter 2.1.

On the other hand, (CH-UC) is not only consistent, but also equivalent to (UC), which can be formulated and proved in intensional MLTT extended with (1) the axiom of function extensionality and (2) a type former $\|-\|$ for propositional truncations. In such extension, (UC) is formulated as

$$\Pi(f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}).\ \|\Sigma(m{:}\mathbb{N}).\ \Pi(\alpha,\beta{:}\mathbf{2}^{\mathbb{N}}).\ \alpha =_m \beta \to f\alpha = f\beta\|.$$

It is not isomorphic to (CH-UC), because in general there can be many uniform-continuity moduli of the same $f\colon \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$. But they are logically equivalent (see Theorem 2.3.1).

The crucial idea is that, by requiring the existence of a *minimal* modulus of uniform continuity, the type of uniform continuity becomes a proposition, with the aid of function extensionality in the proof. Using the elimination rule of $\|-\|$ and the fact that a map $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ is uniformly continuous if and only if it has a minimal modulus of uniform continuity, we get the desired logical equivalence.

Further work 1. Investigate the compatibility of continuity with univalence: Voevodsky's *Univalence Axiom* [72] is also in the heart of viewing types as topological spaces, originally for the purposes of homotopy theory, but more recently with applications much beyond that. It would be interesting to understand the compatibility of continuity with univalence in the context of intensional Martin-Löf type theory, *e.g.* to contribute to the computational understanding of univalence.

9.2 A constructive variation of the topological topos

As discussed in Chapter 3.1, Johnstone's treatment of the topological topos in [48] is non-constructive. To build the topological topos, one starts with the monoid of continuous endomaps of the one-point compactification $\mathbb{N}_\infty$ of the discrete natural numbers, and then takes sheaves for the canonical topology of this monoid considered as a category. In order to obtain an explicit description of the canonical coverage, [48, §3] uses arguments by contradiction or case analysis via excluded middle.

In Chapter 3 we develop a variation of the topological topos within a minimalistic, constructive meta-language. The first difference is that we take the monoid C of uniformly continuous endomaps of the Cantor space $\mathbf{2}^{\mathbb{N}}$. The second one is that we consider a subcanonical one $\mathcal{J}$, consisting of covering families $\{\mathrm{cons}_s\}_{s\in\mathbf{2}^n}$ for all $n \in \mathbb{N}$, where $\mathrm{cons}_s \colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ is the concatenation map $\alpha \mapsto s\alpha$. Our model is the category of sheaves on the site $(\mathrm{C}, \mathcal{J})$. This sheaf topos is suitable for constructively modelling (UC) as illustrated in Chapters 5 and 6, and retains the "topological" character. However, in this thesis we explore only the aspects of the topos that are necessary for the purpose of modelling (UC).

The concrete sheaves in our 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. The idea is to "topologize" a set X by choosing a designated collection of maps $\mathbf{2}^{\mathbb{N}} \to X$, which we call probes. In this thesis, we mainly work with C-spaces as they have sufficient structure, as discussed in Chapter 3.3 and below, to give a computational interpretation of the uniform-continuity principle.

C-spaces form a (locally) cartesian closed category, as proved in Chapter 3.3.2. The constructions are the same as those in the category of sets, with suitable C-topologies. For example, to get products of C-spaces we "C-topologize" the cartesian products, and to get exponentials of C-spaces we "C-topologize" the sets of continuous maps. The category of C-spaces also has a natural numbers object, which is the set of natural numbers equipped with all uniformly continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ as probes. The Yoneda Lemma, specialized to our topos, amounts to that the continuous maps $\mathbf{2}^{\mathbb{N}} \to X$ are in one-to-one correspondence with the probes on X. Thus, any continuous map $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ amounts to a uniformly continuous map. This allows us to define a functional fan: $(\mathbf{2}^{\mathbb{N}} \to \mathbb{N}) \to \mathbb{N}$ which continuously computes moduli of uniform continuity of maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$.

We emphasize that all the above development of C-spaces can be formulated within

intensional MLTT as discussed in Chapter 7, which we formalized in Agda notation (see Chapter 8) and are available at [76].

Further work 2. Find an alternative coverage for the topos: We are also interested in sharper information about uniform continuity, in the sense that a finite part of the input, not necessarily an initial segment, decides the output. For this, we may consider an alternative coverage based on *overwriting* maps

$$\mathrm{overwrite}(n, b)\colon \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}},$$

where $n \in \mathbb{N}$ and $b \in \mathbf{2}$ indicate that the nth bit of the input sequence is to be overwritten to value b. This is also expected to reduce some computations from exponential to linear time, but this is left for future investigation.

9.3 A constructive manifestation of the Kleene–Kreisel continuous functionals

Our C-spaces are analogous to limit spaces which correspond to concrete sheaves in the topological topos. More precisely speaking, limit spaces can be fully embedded in the category of C-spaces, as proved in Chapter 4.2. Any topological space X becomes a limit space by equipping its underlying set with all continuous maps $\mathbb{N}_\infty \to X$ as its convergent sequences, and becomes a C-space by equipping with all continuous maps $\mathbf{2}^{\mathbb{N}} \to X$ as its probes. Thus $\mathbb{N}_\infty$ is a C-space in our topos, and $\mathbf{2}^{\mathbb{N}}$ is a limit space in the topological topos. Given a limit space X, if we take all continuous maps $\mathbf{2}^{\mathbb{N}} \to X$ (in the sense of limit spaces) to be the probes, then X becomes a C-space. This gives the full embedding of limit spaces into C-spaces. Moreover, limit spaces form an exponential ideal of the category of C-spaces, because the embedding has a left adjoint, mapping a C-space X to a limit space by taking the continuous maps $\mathbb{N}_\infty \to X$ (in the sense of C-spaces) as convergent sequences, which preserves finite products.

When restricted to the Kleene–Kreisel objects in the category of limit spaces, the above embedding becomes an equivalence. Hence, the Kleene–Kreisel spaces can be calculated within C-spaces, by starting from the discrete space of natural numbers and iterating products and exponentials. We emphasize that the proof of this fact is non-constructive. For example, the definitions and proofs of the above embedding and its reflector (left adjoint) contain arguments by case analysis via excluded middle. And the proof that the natural numbers objects in the two categories coincide uses an argument by contradiction. But we also emphasize that Chapter 4.2 is the only part of our work that contains non-constructive arguments.

As mentioned in the previous section, our development of C-spaces, including the cartesian closed structure and the fan functional, is constructive. Thus our C-spaces provide a classically equivalent substitute for the traditional manifestations of the Kleene–Kreisel spaces, which constructively validates the uniform-continuity principle, as discussed in the section below.

We also constructively prove an equivalence of the Kleene–Kreisel continuous hierarchy (within C-spaces) and the full type hierarchy, by assuming the Brouwerian principle that all set-theoretic functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous (Chapter 4.3). The crucial but obvious fact is that, under this assumption, the C-space $\mathbb{N}$ becomes indiscrete, i.e. all

maps into $\mathbb{N}$ are continuous. Since the category of indiscrete C-spaces is equivalent to the one of sets, and products and exponentials preserve indiscreteness, we obtain the desired equivalence of simple-type hierarchies. It is interesting that other Brouwerian axioms are not needed in the proof. This result is also related to Fourman's recent work [39] on the principle of predicative reflection.

9.4 Constructive validations of UC in intuitionistic type theories

C-spaces have sufficient structure to model intuitionistic type theories, as mentioned above, in which the uniform-continuity principle (UC) is validated.

Firstly, C-spaces form a cartesian closed category with a natural numbers object $\mathbb{N}$ (Chapter 3.3), and thus give a model of Gödel's system T (Chapter 5.1). Specifically, the term language of system T is modelled as follows: a type is interpreted as a C-space, a context as the product of the interpretations of types in it, and a term in context as a continuous map from the interpretation of its context to the one of its type. Because quantifiers are absent from system T, we introduce a constant $\mathsf{fan}\colon ((\mathsf{N} \to 2) \to \mathsf{N}) \to \mathsf{N}$, and then skolemize the formula of (UC) with the aid of fan as follows:

$$f\colon(\mathsf{N}\to 2)\to\mathsf{N}, \alpha\colon\mathsf{N}\to 2, \beta\colon\mathsf{N}\to 2 \vdash \alpha =_{\mathsf{fan}(f)} \beta \Rightarrow f(\alpha) = f(\beta).$$

Since this additional constant is interpreted as the fan functional, we easily show that the above formula is validated in the model of C-spaces (Theorem 5.1.4), using the definition of the fan functional.

With the same interpretation of the term language of system T, in Chapter 5.2 we define a sort of realizability semantics of HA^ω. We firstly associate to each HA^ω formula ϕ a T type $|\phi|$. In particular, universal quantifications are associated with function types, and existential quantifiers with product types. Then a realizer of ϕ is an element in the C-space which interprets the type $|\phi|$. This time, the principle (UC) can be directly formulated with quantifiers as follows:

$$\vdash \forall f\colon(\mathsf{N}\to 2)\to\mathsf{N}.\ \exists n\colon\mathsf{N}.\ \forall\alpha\colon\mathsf{N}\to 2.\ \forall\beta\colon\mathsf{N}\to 2.\ \alpha =_n \beta \Rightarrow f(\alpha) = f(\beta).$$

Again we use the fan functional to realize the above HA^ω formula (Theorem 5.2.2).

Moreover, the category of C-spaces is locally cartesian closed (Chapter 3.3) and thus also gives a model of dependent types. In Seely's model [66] of (extensional) MLTT in a locally cartesian closed category $\mathbf{C}$, a type $\Gamma \vdash A$ is interpreted as an object, *i.e.* a $\mathbf{C}$-morphism $\bar{A} \to \bar{\Gamma}$, in the slice category $\mathbf{C}/\bar{\Gamma}$, and a term $\Gamma \vdash u : A$ as a section of the interpretation of its type. In particular, Π-types are interpreted by right adjoints to pullback functors, Σ-types by left adjoints to pullback functors, and identity types by equalizers. In the locally cartesian closed category of C-spaces, the Curry–Howard formulation of (UC)

$$\vdash \Pi(f\colon(\mathsf{N}\to 2)\to\mathsf{N}).\ \Sigma(n\colon\mathsf{N}).\ \Pi(\alpha\colon\mathsf{N}\to 2).\ \Pi(\beta\colon\mathsf{N}\to 2).\ \alpha =_n \beta \to f(\alpha) = f(\beta)$$

is interpreted as the unique continuous map $[\![\mathrm{UC}]\!] \to \mathbf{1}$ for some C-space $[\![\mathrm{UC}]\!]$. Then we show that the above type is validated by constructing an inhabitant of $[\![\mathrm{UC}]\!]$ using the

fan functional (Theorem 6.2.1), which is equivalent to saying that the interpretation of the above type has sections.

Seely's interpretation has a well-known coherence issue, caused by interpreting substitutions via pullbacks [25]. Nevertheless, the above proof still makes sense, because locally cartesian closed categories (LCCCs) and categories with families (CwFs) [29], that is a refined notion of Seely's model, are biequivalent [17], and thus a type is inhabited in an LCCC if and only if it is inhabited in the corresponding CwF. Instead of using one direction of the biequivalence, which amounts to Hofmann's construction [42], we directly give a CwF structure to the category of C-spaces in Chapter 6.4. The idea of this model is that each context is (interpreted as) a C-space, while a type on a context Γ consists of a family of sets indexed by the elements of Γ and a family of C-topologies indexed by the probes on Γ. We also give a CwF structure to the supercategory of sheaves in Chapter 6.5, following Coquand's construction of presheaf models [21].

Further work 3. Model UC in the CwF of sheaves: In order to interpret universes, we have to work with the supercategory of sheaves, because universes in sheaf toposes do not correspond to spaces [34]. To validate UC using sheaves, we have to understand the behaviour of a fan functional in the category of sheaves. One possible approach is to apply the inclusion functor C-$\mathbf{Space} \to \mathbf{Shv}(\mathrm{C}, \mathcal{J})$ to the fan functional in the category of C-spaces, and then investigate the resulting natural transformation.

Further work 4. Generalize the method to other principles: As discussed in Chapter 1, sheaf models have been employed to study other non-classical principles, *e.g.* other forms of continuity principles and Bar induction. We are interested in generalizing our methods to investigate such principles in type theory for the purposes of computation.

9.5 Construction of the model in type theory and the Agda formalization

The main aim of the thesis is to extract computational content from type-theoretic proofs that use Brouwer's axiom of uniform continuity. We achieved this by reasoning constructively, as in Chapters 3–6, and by formalizing the development in Martin-Löf type theory (in Agda notation), as in Chapters 7 and 8.

Due to the absence of function extensionality and the presence of proof relevance in MLTT, some issues arise in the formulation of the model construction and in the proofs of the theorems, *e.g.* in the verification of the sheaf condition for exponentials and in the continuity witness of the fan functional.

To overcome the difficulty of proof relevance, we adjusted the notion of uniform continuity, by requiring the existence of a least modulus, to make it a proposition (that is, a type with at most one element). If two uniformly continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are pointwise equal, then they have the same least modulus of uniform continuity and hence yield the identical morphism in the model.

To address the issues caused by the lack of function extensionality (funext), we developed the following approaches:

1. Use setoids. This approach works without requiring extension of type theory. But it gives a long formalization that obscures the essential aspects of the constructions and proofs.

2. Postulate (funext). This simplest approach gives a clean formalization. But it destroys the computational content of formal proofs, which defeats the main aim of the thesis.

3. Postulate (funext) within a computationally irrelevant field. This approach makes use of Agda's irrelevant fields to attempt to verify the conjecture that the usage of (funext) has no computational content. To make it work, an additional probe axiom has to be introduced, which results in a slightly different model. Moreover, it requires the non-standard extension of type theory with such irrelevant fields.

4. Postulate the double negation of (funext). This approach was motivated by the observation that the only property of irrelevant fields we used in the previous approach is that they form a monad T with $T\emptyset \to \emptyset$. Though the model it produces is only *classically* equivalent to the one explained in Chapter 3.3, it provides the same interpretation to simple and dependent types. More importantly, it does not destroy computational content, because the postulated axiom is in a negative form.

In practice, we confine ourselves to a fragment of Agda corresponding to intensional MLTT, with extensions corresponding to the above approaches. Our Agda implementation, which is available at [76], has over 16,000 lines split into 85 modules.

Further work 5. Implement the work in cubical type theory: In addition to the above approaches, we can also develop the model in a variation of MLTT in which (funext) is available. Such a choice could be *cubical type theory* [9] which has a Haskell implementation [19] with a syntax highly close to the one of Agda. Since cubical type theory is an implementation of Voevodsky's univalence axiom, developing our model in it would be also an approach to investigate the consistency of continuity with univalence.

Further work 6. Internalize the syntax and the models of type theory: There are a number of benefits of implementing the syntax and models in type theory, *e.g.* for the purposes of formal verification and of algorithm abstraction. However, as mentioned in Chapter 7.3, this problem remains open in the community of type theory [15, 27, 29, 75].

9.6 Universes in sheaf models

This section discusses preliminary and further work on a notion of universes in sheaf models that is suitable for type-theoretic development.

Following Coquand's construction of the (first) universe in presheaf models [21] (in the sense of Dybjer's categories with families), we attempt to interpret universes in our sheaf model which has been developed in Chapter 6.5. However, the resulting "universe" has two problems regarding its development in Martin-Löf type theory:

1. The underlying type family of the "universe" is not proposition-valued.

   The underlying type of a (pre)sheaf has to be a proposition (*i.e.* a type with at most one element) and hence the underlying type family of a type has to be proposition-valued, when the equations of (pre)sheaves and of types are interpreted using intensional identity types (see Chapter 7.3.2).

2. The amalgamation of any family of elements of the "universe" is unique only up to isomorphism.

The existence of amalgamations is required to be unique up to equality. And the uniqueness is necessary, for instance, when verifying the sheaf condition for exponentials of sheaves.

We remark that the above problems in fact occur in any sheaf topos when performing Coquand's construction.

Recall that a *universe* (à la Tarski [43]) is a type U whose elements are regarded as names of types. Each type can be encoded as an element of U, *i.e.* for each type A, there is an element $|A|\colon \mathrm{U}$. And each element of U can be decoded to a type, *i.e.* for $T\colon \mathrm{U}$, there is a type $\mathrm{El}(T)$. The computation rules of U include

$$A = \mathrm{El}(|A|) \qquad T = |\mathrm{El}(T)|$$

for any type A and any element $T : \mathrm{U}$.

Based on Coquand's presentation of universes in his note [21], we define an additional structure on categories with families (CwFs) to interpret universes. The definition of CwFs and the structures of Σ-types, Π-types and identity types have been introduced in Chapter 6.3. A category with families $\mathbf{C}$ supports the *universe* if and only if

- for any context Γ we have a type $\mathrm{U} \in \mathrm{Type}(\Gamma)$,
- for any type $A \in \mathrm{Type}(\Gamma)$ we have a term $|A| \in \mathrm{Term}(\Gamma, \mathrm{U})$, and
- for any term $T \in \mathrm{Term}(\Gamma, \mathrm{U})$ we have a type $\mathrm{El}(T) \in \mathrm{Type}(\Gamma)$,

such that the following equations hold:

$$\mathrm{El}(|A|) = A \qquad |\mathrm{El}(T)| = T$$

$$\mathrm{U}[\sigma] = \mathrm{U} \qquad |A|[\sigma] = |A[\sigma]| \qquad (\mathrm{El}(T))[\sigma] = \mathrm{El}(t[\sigma]).$$

The corresponding GAT-presentation is given by the following typing rules:

$$\frac{\Gamma \vdash}{\Gamma \vdash \mathrm{U}} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash |A| : \mathrm{U}} \qquad \frac{\Gamma \vdash T : \mathrm{U}}{\Gamma \vdash \mathrm{El}(T)\ \mathrm{Type}}$$

together with the same equations as above.

In Chapter 6.5 we develop a CwF-structure on the category of sheaves on the uniform-continuity site, as well as the additional structures to interpret Σ- and Π-types, following Coquand's construction of presheaf model, within a meta-language of some form of constructive set theory. Here we attempt to extend his construction to interpret the (first) universe.

We firstly construct a (large) set U^* as follows: An element $A \in \mathrm{U}^*$ is a C-indexed family of sets together with restriction maps

$$\_ \bullet \_ : \ A_t \to \prod_{r:\mathrm{C}} A_{(t \circ r)}$$

for each $t\colon \mathrm{C}$ such that

(u1) $a \bullet 1 = a$ for all $t \in \mathrm{C}$ and $a \in A_t$,

(u2) $(a \bullet r) \bullet r' = a \bullet (r \circ r')$ for all $t, r, r' \in \mathrm{C}$ and $a \in A_t$,

(u3) for any $t \in \mathrm{C}$, $a_0 \in A_{t \circ \mathrm{cons}_0}$ and $a_1 \in A_{t \circ \mathrm{cons}_1}$, there is a unique amalgamation $a \in A_t$ such that $a \bullet \mathrm{cons}_0 = a_0$ and $a \bullet \mathrm{cons}_1 = a_1$.

Similarly to (s3) and (t3) in Chapter 6.5, we have the following logically equivalent condition of (u3):

(u3') for any $t \in \mathrm{C}$, $n \in \mathbb{N}$ and $\{a_s \in A_{t \circ \mathrm{cons}_s}\}_{s \in \mathbf{2}^n}$, there is a unique amalgamation $a \in A_t$ such that $a \bullet \mathrm{cons}_s = a_s$ for each $s \in \mathbf{2}^n$.

Then we define the *universe* $\Gamma \vdash \mathrm{U}$ to be the constant family consisting of U^*, *i.e.*

$$\mathrm{U}_\gamma \ :\equiv \ \mathrm{U}^*$$

for all $\gamma \in \Gamma$, with the restriction map $\_ * \_ \colon \mathrm{U}^* \to \mathrm{C} \to \mathrm{U}^*$ defined by

$$(A * t)_r \ :\equiv \ A_{t \circ r}$$

for $A \in \mathrm{U}^*$ and $t, r \in \mathrm{C}$. It is routine to prove that $A * t$ satisfies (u1) to (u3), which illustrates that the above restriction map is well-defined.

To be a type, the family U has to satisfy (t1) to (t3). The first two conditions are clearly satisfied. For (t3) we only have the following:

Proposition 9.6.1. *For any $A^0, A^1 \in \mathrm{U}^*$, there exists $A \in \mathrm{U}^*$ such that $A * \mathrm{cons}_i = A^i$ for each $i \in \mathbf{2}$. And A is unique up to pointwise* isomorphism.

Proof. Given $A^0, A^1 \in \mathrm{U}^*$, we construct $A \in \mathrm{U}^*$ as follows: Given $t \in \mathrm{C}$, let $n_t = \mathrm{lmod}_t(1)$. By (†), for each $s \in \mathbf{2}^{n_t}$, there are $i_s \in \mathbf{2}$ and $t_s \in \mathrm{C}$ such that $t \circ \mathrm{cons}_s = \mathrm{cons}_{i_s} \circ t_s$. We define $A_t :\equiv \prod_{s : \mathbf{2}^{n_t}} A^{i_s}_{t_s}$. Given $t, r \in \mathrm{C}$ and $w \in A_t$, we define $w \bullet r \in A_{(t \circ r)}$ as follows: Let $n_t = \mathrm{lmod}_t(1)$ and let $n_r = \mathrm{lmod}_r(n_t)$. Then, using (†) twice, we have, for each $s \in \mathbf{2}^{n_r}$,

$$t \circ r \circ \mathrm{cons}_s = t \circ \mathrm{cons}_{s_s} \circ r_s = \mathrm{cons}_{i_s} \circ t_s \circ r_s \qquad (\dagger\dagger)$$

for some $s_s \in \mathbf{2}^{n_t}$, $t_s, r_s \in \mathrm{C}$ and $i_s \in \mathbf{2}$. Let $n = \mathrm{lmod}_{t \circ r}(1)$. For each $s \in \mathbf{2}^n$, we have $i_s \in \mathbf{2}$ and $t_s \in \mathrm{C}$ such that $t \circ r \circ \mathrm{cons}_s = \mathrm{cons}_{i_s} \circ t_s$. Since $n \leq n_r$, for each $s' \in \mathbf{2}^{(n_r - n)}$, we have

$$t \circ r \circ \mathrm{cons}_s \circ \mathrm{cons}_{s'} = \mathrm{cons}_{i_s} \circ t_s \circ \mathrm{cons}_{s'}$$

and, using (††), we have $s_{ss'} \in \mathbf{2}^{n_t}$, $i_{ss'} \in \mathbf{2}$ and $t_{ss'}, r_{ss'} \in \mathrm{C}$ such that

$$t \circ r \circ \mathrm{cons}_s \circ \mathrm{cons}_{s'} = t \circ r \circ \mathrm{cons}_{ss'} = \mathrm{cons}_{i_{ss'}} \circ t_{ss'} \circ r_{ss'}.$$

It is not difficult to prove that $i_s = i_{ss'}$, and thus $t_s \circ \mathrm{cons}_{s'} = t_{ss'} \circ r_{ss'}$. Therefore, for each $s \in \mathbf{2}^n$, we have a family $\{w(s_{ss'}) \bullet r_{ss'} \in A^{i_s}_{t_s \circ \mathrm{cons}_{s'}}\}_{s' \in \mathbf{2}^{(n_r - n)}}$. By (u3') of A^{i_s}, this family has a unique amalgamation $a_s \in A^{i_s}_{t_s}$. We define $(w \bullet r)(s) :\equiv a_s$ for each $s \in \mathbf{2}^n$. Then we show that A satisfies (u1) to (u3):

1. Given $t \in \mathrm{C}$ and $w \in A_t$, we have $w \bullet 1 = w$ because $\mathrm{lmod}_t(1) = \mathrm{lmod}_{(t \circ 1)}(1)$.

2. Given $t, r, r' \in \mathrm{C}$ and $w \in A_t$, it is routine to prove $(w \bullet r) \bullet r' = w \bullet (r \circ r')$ following the definition of $\_ \bullet \_$ on A and using the uniqueness of amalgamations.

3. Given $t \in \mathrm{C}$, $w_0 \in A_{t \circ \mathrm{cons}_0}$ and $w_1 \in A_{t \circ \mathrm{cons}_1}$, we define $w \in A_t$ as follows: Let $n_t = \mathrm{lmod}_t(1)$. (i) If $n_t = 0$ then $t = \mathrm{cons}_i \circ t'$ for some $i \in \mathbf{2}$ and $t' \in \mathrm{C}$. Moreover, we know $\mathrm{lmod}_{(t \circ \mathrm{cons}_0)} = \mathrm{lmod}_{(t \circ \mathrm{cons}_1)} = 0$, and thus $w_0(\varepsilon) \in A^i_{t' \circ \mathrm{cons}_0}$ and $w_1(\varepsilon) \in A^i_{t' \circ \mathrm{cons}_1}$. We define $w(\varepsilon)$ to be the unique amalgamation of $w_0(\varepsilon)$ and $w_1(\varepsilon)$, by (u3) of A^i. (ii) If $n_t = n + 1$ for some n, then $n \geq n_0 \equiv \mathrm{lmod}_{(t \circ \mathrm{cons}_0)}(1)$ and $n \geq n_1 \equiv \mathrm{lmod}_{(t \circ \mathrm{cons}_1)}(1)$. For each $i \in \mathbf{2}$ and $s \in \mathbf{2}^n$, we define $w(is) :\equiv w_i(s_0) \bullet \mathrm{cons}_{s_1}$ where $s_0 \in \mathbf{2}^{n_i}$ and $s_1 \in \mathbf{2}^{n - n_i}$ such that $s = s_0 s_1$. Then it is routine to check that $w \bullet \mathrm{cons}_0 = w_0$ and $w \bullet \mathrm{cons}_1 = w_1$ hold pointwise in both (i) and (ii).

Therefore, A is an element of U^*.

The above A is an "amalgamation" of A^0 and A^1: for each $i \in \mathbf{2}$, we have

$$\begin{array}{rcll} (A * \mathrm{cons}_i)_t & = & A_{\mathrm{cons}_i \circ t} & \text{(by the definition of } \_ * \_ \text{ on U)} \\ & = & \prod_{s \in \mathbf{2}^0} A^i_t & \text{(by the definition of } A \text{ and the fact } \mathrm{lmod}_{(\mathrm{cons}_i \circ t)}(1) = 0) \\ & = & A^i_t & \end{array}$$

for any $t \in \mathrm{C}$.

Finally we show that A is unique up to isomorphism: Suppose we are given $B \in \mathrm{U}^*$ such that $B * \mathrm{cons}_i = A^i$ for each $i \in \mathbf{2}$. Given $t \in \mathrm{C}$, we let $n = \mathrm{lmod}_t(1)$ and for each $s \in \mathbf{2}^n$ have $t \circ \mathrm{cons}_s = \mathrm{cons}_{i_s} \circ t_s$ for some $i_s \in \mathbf{2}$ and $t_s \in \mathrm{C}$. We have $B_t \cong A_t$:

($\Rightarrow$) Let $b \in B_t$. For each $s \in \mathbf{2}^n$ we have $b \bullet \mathrm{cons}_s \in B_{t \circ \mathrm{cons}_s}$. Because $B_{t \circ \mathrm{cons}_s} = A^{i_s}_{t_s}$ the tuple consisting of all $b \bullet \mathrm{cons}_s$ is an element of A_t.

($\Leftarrow$) Let $w \in A_t$. For each $s \in \mathbf{2}^n$ we have $w(s) \in A^{i_s}_{t_s}$. Because $B_{t \circ \mathrm{cons}_s} = A^{i_s}_{t_s}$ we have a family $\{w(s) \in B_{t \circ \mathrm{cons}_s}\}_{s \in \mathbf{2}^n}$. Using (u3') of B, it has a unique amalgamation in B_t.

Clearly the composites of the above operations are identity. □

Suppose that, in the meta-theory, isomorphisms of sets yield equalities. Then the above proof does construct amalgamations of elements in U^*. We carry on the construction of the first universe, including the coding operator $|-|$ and the decoding operator El, in our sheaf model. Given a type $\Gamma \vdash A$, we define a term $\Gamma \vdash |A| : \mathrm{U}$ by

$$|A| \colon \prod_{\gamma : \Gamma} \mathrm{U}_\gamma \qquad (|A|(\gamma))_t :\equiv A_{\gamma \cdot t}$$

and the map $\_ \bullet \_$ is defined by, for any $a \in (|A|(\gamma))_t$ and $r \colon \mathrm{C}$,

$$a \bullet r \;:\equiv\; a * r,$$

where $\_ * \_$ is the restriction map of A. We skip the routine proof that $|A|$ is well defined. Given a term $\Gamma \vdash T : \mathrm{U}$, we define a type $\Gamma \vdash \mathrm{El}(T)$ by $(\mathrm{El}(T))_\gamma :\equiv (T(\gamma))_1$ for each $\gamma \in \Gamma$. The restriction maps of $\mathrm{El}(T)$ are defined by, for $\gamma \in \Gamma$, $a \in (\mathrm{El}(T))_\gamma$ and $t \in \mathrm{C}$,

$$a * t \;:\equiv\; a \bullet t,$$

where $\_ \bullet \_$ is the restriction map on $T(\gamma)$. It is routine to prove that $\mathrm{El}(T)$ is a well defined term. It remains to verify the required equations as presented above, which is also routine.

Further work 7. Refine the notion of universe in sheaf models: As shown above, when extending Coquand's construction of the universe in presheaf models to the one in our (or any) sheaf model, the amalgamation of any family of elements of the "universe" is unique only up to isomorphism. Streicher [70] constructs universes in a presheaf topos and then sheafifiies them to get the ones in sheaves. However, his construction, *e.g.* sheafification, does not seem suitable for development in predicative intuitionistic type theories. It remains open to develop a notion of universes in sheaf toposes that is suitable for type-theoretic development.

Bibliography

[1] Agda Community. The Agda Wiki – The Agda reference manual. Available at: http://wiki.portal.chalmers.se/agda/ReferenceManual.TOC.

[2] S. Awodey and A. Bauer. Propositions as [types]. *Journal of Logic and Computation*, 14(4):447–471, 2004.

[3] J. C. Baez and A. E. Hoffnung. Convenient categories of smooth spaces. *Transactions of the American Mathematical Society*, 363(11):5789–5825, 2011.

[4] I. Battenfeld, M. Schröder, and A. Simpson. Compactly generated domain theory. *Mathematical Structures in Computer Science*, 16(2):141–161, 2006.

[5] A. Bauer, L. Birkedal, and D. S. Scott. Equilogical spaces. *Theoretical Computer Science*, 315(1):35–59, 2004.

[6] A. Bauer and A. Simpson. Continuity begets continuity. Presented at *Trends in constructive mathematics* in Germany, 2006.

[7] M. J. Beeson. *Foundations of Constructive Mathematics*. Springer-Verlag, 1985.

[8] Y. Bertot, P. Castéran, G. Huet, and C. Paulin-Mohring. *Interactive theorem proving and program development : Coq'Art : the calculus of inductive constructions*. Texts in theoretical computer science. Springer, Berlin, New York, 2004.

[9] M. Bezem, T. Coquand, and S. Huber. A model of type theory in cubical sets. Preprint, March 2014.

[10] E. Bishop. *Foundations of Constructive Analysis*. McGraw-Hill, 1967.

[11] A. Bove and P. Dybjer. Dependent types at work. Lecture Notes for the LerNet Summer School, Piriápolis, Uruguay, 2008.

[12] A. Bove, P. Dybjer, and U. Norell. A brief overview of Agda—a functional language with dependent types. In *Theorem proving in higher order logics*, volume 5674 of *Lecture Notes in Computer Science*, pages 73–78. Springer, 2009.

[13] D. Bridges and F. Richman. *Varieties of Constructive Mathematics*. Cambridge University Press, 1987.

[14] J. Cartmell. Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32(0):209–243, 1986.

[15] J. Chapman. Type theory should eat itself. *Electronic Notes in Theoretical Computer Science*, 228(0):21–36, 2009.

[16] P. Clairambault. From categories with families to locally cartesian closed categories. M1 report on categorical models of type theory, 2006.

[17] P. Clairambault and P. Dybjer. The biequivalence of locally cartesian closed categories and martin-löf type theories. In *Typed Lambda Calculi and Applications*, volume 6690 of *Lecture Notes in Computer Science*, pages 91–106. Springer Berlin Heidelberg, 2011.

[18] J. Cockx, D. Devriese, and F. Piessens. Pattern matching without k. In *Proceedings of the 19th ACM SIGPLAN International Conference on Functional Programming*, pages 257–268. ACM, 2014.

[19] C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. Implementation of univalence in cubical sets. Available at `https://github.com/simhu/cubical`.

[20] T. Coquand. Pattern matching with dependent types. In *Proceedings of the Workshop on Types for Proofs and Programs*, pages 66–79, 1992.

[21] T. Coquand. Sheaf model of type theory. Unpulished note, 2013.

[22] T. Coquand, N. A. Danielsson, M. H. Escardó, U. Norell, and C. Xu. Negative consistent axioms can be postulated without loss of canonicity. Unpublished note, 2013.

[23] T. Coquand and G. Jaber. A note on forcing and type theory. *Fundamenta Informaticae*, 100(1-4):43–52, 2010.

[24] T. Coquand and G. Jaber. A computational interpretation of forcing in type theory. In *Epistemology versus Ontology*, volume 27, pages 203–213. Springer Netherlands, 2012.

[25] P.-L. Curien. Substitution up to isomorphism. *Fundamenta Informaticae*, 19(1/2):51–85, 1993.

[26] P.-L. Curien, R. Garner, and M. Hofmann. Revisiting the categorical interpretation of dependent type theory. *Theoretical Computer Science*, 546(0):99–119, 2014.

[27] N. Danielsson. A formalisation of a dependently typed language as an inductive-recursive family. In *Types for Proofs and Programs*, volume 4502 of *Lecture Notes in Computer Science*, pages 93–109. Springer Berlin Heidelberg, 2007.

[28] E. Dubuc. Concrete quasitopoi. In *Applications of Sheaves*, volume 753 of *Lecture Notes in Mathematics*, pages 239–254. Springer Berlin / Heidelberg, 1979.

[29] P. Dybjer. Internal type theory. In *Types for proofs and programs (Torino, 1995)*, volume 1158 of *Lecture Notes in Computer Science*, pages 120–134. Springer, 1996.

[30] M. H. Escardó. Synthetic topology of data types and classical spaces. *Electron. Notes Theor. Comput. Sci.*, 87:21–156, 2004.

[31] M. H. Escardó. In intensional Martin-Löf type theory, if all functions (n-¿n)-¿n are continuous then 0=1. Note with an informal proof and with formal proof in Agda, 2013.

[32] M. H. Escardó. Infinite sets that satisfy the principle of omniscience in any variety of constructive mathematics. *Journal of Symbolic Logic*, 78(3):764–784, 2013.

[33] M. H. Escardó, J. Lawson, and A. Simpson. Comparing Cartesian closed categories of (core) compactly generated spaces. *Topology and its Applications*, 143(1-3):105–145, 2004.

[34] M. H. Escardó and T. Streicher. The universe is indiscrete. Submitted for publication, 2013.

[35] M. H. Escardó and C. Xu. A constructive manifestation of the Kleene–Kreisel continuous functionals. Submitted for publication, 2013.

[36] M. H. Escardó and C. Xu. The inconsistency of a Brouwerian continuity principle with the Curry–Howard interpretation. Submitted for publication, 2015.

[37] M. P. Fourman. Notions of choice sequence. In *The L. E. J. Brouwer Centenary Symposium Proceedings of the Conference held in Noordwijkerhout*, volume 110, pages 91–105, 1982.

[38] M. P. Fourman. Continuous truth I, non-constructive objects. In *Proceedings of Logic Colloquium, Florence 1982*, volume 112, pages 161–180. Elsevier, 1984.

[39] M. P. Fourman. Continuous truth II: reflections. In *Workshop on Logic, Language, Information and Computation*, volume 8071 of *Lecture Notes in Computer Science*, pages 153–167. Springer Berlin Heidelberg, 2013.

[40] M. Hedberg. A coherence theorem for Martin-Löf's type theory. *J. Functional Programming*, pages 413–436, 1998.

[41] M. Hofmann. *Extensional concepts in intensional type theory*. PhD thesis, University of Edinburgh, 1995. Technical report ECS-LFCS-95-327.

[42] M. Hofmann. On the interpretation of type theory in locally cartesian closed categories. In *Computer science logic (Kazimierz, 1994)*, volume 933 of *Lecture Notes in Computer Science*, pages 427–441. Springer, 1995.

[43] M. Hofmann. Syntax and semantics of dependent types. In *Semantics and Logics of Computation*, pages 79–130. Cambridge University Press, 1997.

[44] M. Hofmann and T. Streicher. The groupoid interpretation of type theory. In *In Venice Festschrift*, pages 83–111. Oxford University Press, 1996.

[45] J. M. E. Hyland. Filter spaces and continuous functionals. *Annals of Mathematical Logic*, 16:101–143, 1979.

[46] J. M. E. Hyland. The effective topos. In *The L.E.J. Brouwer Centenary Symposium (Noordwijkerhout, 1981)*, volume 110 of *Studies in Logic and the Foundations of Mathematics*, pages 165–216. North-Holland, 1982.

[47] M. Hyland. Function spaces in the category of locales. In *Continuous lattices*, volume 871 of *Lect. Notes Math.*, pages 264–281, 1981.

[48] P. T. Johnstone. On a topological topos. *Proceedings of the London Mathematical Society*, 38(3):237–271, 1979.

[49] P. T. Johnstone. *Sketches of an elephant: a topos theory compendium. Vol. 1*, volume 43 of *Oxford Logic Guides*. The Clarendon Press Oxford University Press, 2002.

[50] S. C. Kleene. Countable functionals. In A. Heyting, editor, *Constructivity in Mathematics*, pages 81–100. North-Holland, 1959.

[51] N. Kraus, M. Escardó, T. Coquand, and T. Altenkirch. Generalizations of hedberg's theorem. In *Typed Lambda Calculi and Applications*, volume 7941 of *Lecture Notes in Computer Science*, pages 173–188. Springer Berlin Heidelberg, 2013.

[52] N. Kraus, M. Escardó, T. Coquand, and T. Altenkirch. Notions of anonymous existence in Martin-Löf Type Theory. Submitted for publication, 2014.

[53] G. Kreisel. Interpretation of analysis by means of constructive functionals of finite types. In A. Heyting, editor, *Constructivity in Mathematics*, pages 101–128. North-Holland, 1959.

[54] G. Kreisel. On weak completeness of intuitionistic predicate logic. *J. Symbolic Logic*, 27:139–158, 1962.

[55] J. Lambek and P. J. Scott. *Introduction to Higher-Order Categorical Logic*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1988.

[56] J. R. Longley. Notions of computability at higher types. I. In *Logic Colloquium 2000*, volume 19 of *Lecture Notes in Logic*, pages 32–142. The Association for Symbolic Logic, 2005.

[57] J. R. Longley. On the ubiquity of certain total type structures. *Mathematical Structures in Computer Science*, 17(5):841–953, 2007.

[58] S. Mac Lane. *Categories for the Working Mathematician*. Springer, 2nd edition, 1998.

[59] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer, 1992.

[60] P. Martin-Löf. An intuitionistic theory of types: Predicative part. In H. E. Rose and S. J. C., editors, *Logic Colloquium 1973*, pages 73–118. North-Holland, 1975.

[61] P. Martin-Löf. *Intuitionistic type theory*. Studies in proof theory. Bibliopolis, 1984. Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980.

[62] M. Menni and A. Simpson. Topological and limit-space subcategories of countably-based equilogical spaces. *Mathematical Structures in Computer Science*, 12:739–770, 2002.

[63] U. Norell. Dependently typed programming in Agda. In *Proceedings of the 4th international workshop on Types in language design and implementation*, TLDI '09, pages 1–2, 2009.

[64] D. Normann. *Recursion on the countable functionals*, volume 811 of *Lecture Notes in Mathematics*. Springer, 1980.

[65] D. Normann. Computing with functionals - computability theory or computer science? *Bulletin of Symbolic Logic*, 12(1):43–59, 2006.

[66] R. A. G. Seely. Locally cartesian closed categories and type theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 95(1):33–48, 1984.

[67] G. F. Simmons. *Introduction to Topology and Modern Analysis*. Pure and applied mathematics. McGraw-Hill, 1963.

[68] E. H. Spanier. Quasi-topologies. *Duke Mathematical Journal*, 30(1):1–14, 1963.

[69] T. Streicher. Investigations into intensional type theory. Habilitationsschrift, LMU München, 1993.

[70] T. Streicher. Universes in toposes. In *From sets and types to topology and analysis*, volume 48 of *Oxford Logic Guides*, pages 78–90. Oxford University Press, 2005.

[71] M. B. Symth. Topology. In S. Abramsky, D. M. Babbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume I, pages 641–761. Clarendon Press, 1993.

[72] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013.

[73] A. S. Troelstra and D. van Dalen. *Constructivism in Mathematics : An Introduction*. North-Holland, 1988.

[74] G. van der Hoeven and I. Moerdijk. Sheaf models for choice sequences. *Annals of Pure and Applied Logic*, 27(1):63–107, 1984.

[75] V. Voevodsky. HoTT is not an interpretation of MLTT into abstract homotopy theory. Post available at http://homotopytypetheory.org/2015/01/11/hott-is-not-an-interpretation-of-mltt-into-abstract-homotopy-theory/, 2005.

[76] C. Xu. A continuous computational interpretation of type theories, developed in Agda. Available at http://cj-xu.github.io/ContinuityType/, 2015.

[77] C. Xu and M. H. Escardó. A constructive model of uniform continuity. In *Typed Lambda Calculi and Applications*, volume 7941 of *Lecture Notes in Computer Science*, pages 236–249. Springer Berlin Heidelberg, 2013.

Index