

A SYNTACTIC APPROACH TO CONTINUITY OF T-DEFINABLE FUNCTIONALS

CHUANGJIE XU

Ludwig-Maximilians-Universität München
e-mail address: xu@math.lmu.de

ABSTRACT. We give a *new* proof of the well-known fact that all functions $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ which are definable in Gödel’s System T are continuous via a *syntactic* approach. Differing from the usual syntactic method, we firstly perform a translation of System T into itself in which natural numbers are translated to functions $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$. Then we inductively define a continuity predicate on the translated elements and show that the translation of any term in System T satisfies the continuity predicate. We obtain the desired result by relating terms and their translations via a parametrized logical relation. Our constructions and proofs have been formalized in the Agda proof assistant. Because Agda is also a programming language, we can execute our proof to compute moduli of continuity of T-definable functions.

1. INTRODUCTION

The usual syntactic method for proving properties of terms in Gödel’s System T works as follows: (1) Define a predicate $P_\rho \subseteq \rho$ for the designated property by induction on the finite type ρ , *i.e.*

$$\begin{aligned} P_{\mathbb{N}}(n) &::= \dots \\ P_{\sigma \rightarrow \tau}(g) &::= \forall x^\sigma (P_\sigma(x) \rightarrow P_\tau(g(x))). \end{aligned}$$

(2) Prove that each term $t : \rho$ in T satisfies P_ρ by induction on t . Examples of properties for which such syntactic methods work include totality [18] and majorizability [11]. Our goal is to recover the well-known fact that all T-definable functions $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ are continuous [2]. But the above syntactic method does not seem to work directly, because we do not know how to define the base case $P_{\mathbb{N}}$ such that $P_{(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}}(f)$ expresses the continuity of f .

Our idea is, as a step (0), to perform a translation $(t \mapsto t^b) : \rho \rightarrow \rho^b$ of System T into itself where $\mathbb{N}^b ::= (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$, *i.e.* natural numbers are translated to functions $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$. Further, any function $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ in T is pointwise equal to $f^b(\Omega)$, where $f^b : (\mathbb{N}^b \rightarrow \mathbb{N}^b) \rightarrow \mathbb{N}^b$ is the translation of f and $\Omega : \mathbb{N}^b \rightarrow \mathbb{N}^b$ is a T-definable generic element in the sense of [3, 4, 5]. Then our step (1) is to inductively define a continuity predicate $C_\rho \subseteq \rho^b$: the base case $C_{\mathbb{N}}(f)$ states the continuity of $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ and the one for function spaces is defined in the usual way as above. And our step (2) is to

Key words and phrases: syntactic method, Gödel’s System T, (uniform) continuity, Baire space, Cantor space, logical relation, Agda.

prove $C_\rho(t^b)$ for all terms $t : \rho$ in \mathbb{T} by induction on t . Moreover, the generic element Ω also satisfies the continuity predicate C . We thus have $C_{\mathbb{N}}(f^b(\Omega))$, *i.e.* $f^b(\Omega)$ is continuous, for any $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ in \mathbb{T} . Because continuity is preserved under pointwise equality, we know that f is also continuous. As pointed out by one of the anonymous referees, our results that any term is related to its translation (Lemma 2.3) and that the translation of any term satisfies the continuity predicate (Lemma 3.2) are instances of the *fundamental theorem of logical relations* [19].

Our development is constructive and has been formalized in the Agda proof assistant [1]. The main purpose of this formalization is to execute our Agda proof which is also a computer program to compute moduli of continuity, rather than merely certify the correctness of our work. Some sample computations of moduli of continuity are provided in our Agda development [25].

Motivation. Oliva and Steila [13] give a direct proof of Schwichtenberg’s theorem that the terms of Gödel’s System \mathbb{T} are closed under the rule of Spector’s bar recursion of types 0 and 1 [16]. By induction on terms, they explicitly construct a functional of their notion of *general bar recursion* for each term in \mathbb{T} , and then turn it into a functional of Spector’s bar recursion. The author implemented their work in Agda [24] and recognized that some part of their correctness proof [13, Theorem 3.4] can be adapted to compute moduli of continuity. For each term t in \mathbb{T} , a bar is constructed to control the behavior of the general-bar-recursion functional for t . This bar essentially contains the continuity information of t . The Agda code of this part of the correctness proof was then separated, refined and further developed under the inspiration of Escardó’s Agda development of dialogue trees [5], which led to the first version of the Agda implementation of this paper [25].

Related work. Kohlenbach [10] obtains the uniform continuity of terms $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ in System \mathbb{T} as an application of his pointwise version of strong majorization. In particular, his logical relation for majorization on functions from natural numbers is given in a pointwise way. He shows that every term $t : \rho$ in \mathbb{T} is pointwise strongly majorized by some $t^* : \rho$ in \mathbb{T} by induction on t . For any $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ in \mathbb{T} , one can show in the intuitionistic system WE-HA^ω that it is extensional. From such a proof, Kohlenbach extracts a term via the Dialectica interpretation and then uses the majorant of this term to construct a modulus of uniform continuity of f .

Coquand and Jaber’s approach to continuity [3, 4] is also syntactic. The difference is that they obtain continuity information by an operational method: they extend dependent type theory with a new constant f for a generic element, and then decorate its operational semantics with forcing information to keep track of approximation information about f as the computations proceed. The continuity information of a function F is extracted from the computation of $F(f)$. They tackle uniform continuity of functions from the Cantor space $\mathbb{N} \rightarrow \mathbb{2}$, where $\mathbb{2} := \{0, 1\}$, and also discuss how to adapt their argument for continuity of functions from the Baire space $\mathbb{N} \rightarrow \mathbb{N}$ [4, Section 1.2.3]. They also provide a Haskell implementation for System \mathbb{T} using a monad combined by the list monad and the state monad as an appendix in [4]. The algorithm to extract continuity information in their operational method restricted to System \mathbb{T} can be represented as a monadic translation of System \mathbb{T} into itself which is an instance of [14, Section 4].

Inspired by Coquand and Jaber’s work, Escardó also employs a generic element but in his dialogue tree model to prove (uniform) continuity of T-definable functions [5]. He has a concrete notion of generic element given by a function $\tilde{\mathbb{N}} \rightarrow \tilde{\mathbb{N}}$, where $\tilde{\mathbb{N}}$ is the set of dialogue trees. Suppose a function $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ is denoted by a term t whose dialogue interpretation is a function $\tilde{f} : (\tilde{\mathbb{N}} \rightarrow \tilde{\mathbb{N}}) \rightarrow \tilde{\mathbb{N}}$. Applying \tilde{f} to the generic element, a dialogue tree which contains the (uniform) continuity information of f is obtained. In one version of the Agda implementation of [5], Escardó uses Church encodings of dialogue trees to turn his semantic interpretation into a compositional translation of System T into itself. In this way, he extracts from the Church encoding of the dialogue tree interpreting t a term m in T which internalizes the modulus of continuity of f . We provide a direct and explicit construction of terms in T internalizing moduli of continuity in Section 4.1 .

Our method is also related to the sheaf model [26, 6, 22] introduced by Escardó and the author. To prove uniform continuity of T-definable functions, it is sufficient to work with the subcategory of concrete sheaves which admit a more intuitive description as what we call C-spaces. A C-space is a set X equipped with a certain collection of maps $\mathbb{2}^{\mathbb{N}} \rightarrow X$ which are called probes on X , where $\mathbb{2}^{\mathbb{N}} := \mathbb{N} \rightarrow \mathbb{2}$. A C-continuous map of C-spaces is a function whose composition with a probe is again a probe. For instance, the set \mathbb{N} with all uniformly continuous maps $\mathbb{2}^{\mathbb{N}} \rightarrow \mathbb{N}$ forms a C-space; so does $\mathbb{2}^{\mathbb{N}}$ with all uniformly continuous maps $\mathbb{2}^{\mathbb{N}} \rightarrow \mathbb{2}^{\mathbb{N}}$. A crucial tool in this semantic method is the Yoneda Lemma which says that a function $\mathbb{2}^{\mathbb{N}} \rightarrow X$ into a C-space X is C-continuous iff it is a probe on X . Each term $t : (\mathbb{N} \rightarrow \mathbb{2}) \rightarrow \mathbb{N}$ in T is interpreted as a C-continuous map $\mathbb{2}^{\mathbb{N}} \rightarrow \mathbb{N}$ in the category of C-spaces and thus, by the Yoneda Lemma, a probe on \mathbb{N} , *i.e.* a uniformly continuous function. Interestingly, the proof of one direction of the Yoneda Lemma is essentially the same proof of that Ω satisfies a certain uniform-continuity predicate studied in Section 4.2. To investigate the relationship between our syntactic method and sheaf models [7, 8, 9, 21] is left as one of our future tasks.

Another different approach to continuity is to use computational effects such as exceptions [12, 15]. Suppose a function $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ and a sequence $\alpha : \mathbb{N} \rightarrow \mathbb{N}$ are given. We can find a number m such that the value of $f(\alpha)$ depends only on the first m positions of α (*i.e.* a modulus of continuity of f at α) as follows: An exception is thrown if f attempts to compute $\alpha(n)$ for $n \geq k$ where k is a variable parameter. We start with computing $f(\alpha)$ with $k = 0$. Once an exception is caught, we try $k + 1$. At some point no exception happens and the current value of k is a modulus of continuity of f .

Organization. Section 2 introduces the b-translation of System T as a preliminary for the syntactic method. Section 3 employs the syntactic method on the b-translation of System T to prove continuity of T-definable functions $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$. Section 4 strengthens the result by constructing terms in System T which internalize moduli of continuity, and studies uniform continuity of T-definable functions $(\mathbb{N} \rightarrow \mathbb{2}) \rightarrow \mathbb{N}$. The last section discusses how to generalize the method for proving properties of T-definable functions of arbitrary finite types.

2. GÖDEL’S SYSTEM T AND THE b-TRANSLATION

We work with Gödel’s System T in its lambda-calculus form. Recall that the term language of T is (equivalent to) a simply typed lambda calculus extended with natural numbers and

a primitive recursor. The constants and equations (*i.e.* computational rules) associated to the ground type \mathbb{N} include

- the natural number $0 : \mathbb{N}$,
- the successor function $\text{succ} : \mathbb{N} \rightarrow \mathbb{N}$, and
- the primitive recursor $\text{rec} : \rho \rightarrow (\mathbb{N} \rightarrow \rho \rightarrow \rho) \rightarrow \mathbb{N} \rightarrow \rho$ with

$$\text{rec}(a)(f)(0) = a \quad \text{rec}(a)(f)(\text{succ } n) = f(n)(\text{rec}(a)(f)(n))$$

for every finite type ρ .

A function is called *T-definable* if there exists a closed term in \mathbb{T} denoting it. In the paper, we do not distinguish T-definable functions and their corresponding T-terms. Moreover, we may write $n + 1$ rather than $\text{succ}(n)$, and α_i rather than $\alpha(i)$ for $\alpha : \mathbb{N} \rightarrow \mathbb{N}$ and $i : \mathbb{N}$.

As discussed earlier, it does not seem possible to directly apply the syntactic method to prove continuity of T-definable functions $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$. Hence we ‘precook’ T-terms so that continuity becomes the base case of a predicate which all ‘precooked’ terms will satisfy. We call this procedure the *b-translation* where *b* stands for the Baire type/space, because natural numbers are translated to functionals from the Baire type $\mathbb{N} \rightarrow \mathbb{N}$.

Definition 2.1 (b-translation). For each finite type ρ we associate a finite type ρ^b inductively as follows:

$$\begin{aligned} \mathbb{N}^b &::= (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N} \\ (\sigma \rightarrow \tau)^b &::= \sigma^b \rightarrow \tau^b. \end{aligned}$$

Assume a given mapping of variables $x : \rho$ to variables $x^b : \rho^b$. For any term $t : \rho$ in \mathbb{T} , we define $t^b : \rho^b$ inductively as follows:

$$\begin{aligned} (x)^b &::= x^b \\ (\lambda x.u)^b &::= \lambda x^b.u^b \\ (fa)^b &::= f^b a^b \\ 0^b &::= \lambda \alpha.0 \\ \text{succ}^b &::= \lambda f \alpha.\text{succ}(f \alpha) \\ \text{rec}^b &::= \lambda a f.\text{ke}(\text{rec}(a)(\lambda k.f(\lambda \alpha.k))) \end{aligned}$$

where $\text{ke}_\rho : (\mathbb{N} \rightarrow \rho^b) \rightarrow \mathbb{N}^b \rightarrow \rho^b$ is inductively defined by

$$\begin{aligned} \text{ke}_{\mathbb{N}}(g)(f) &::= \lambda \alpha.g(f \alpha)(\alpha) \\ \text{ke}_{\sigma \rightarrow \tau}(g)(f) &::= \lambda x.\text{ke}_\tau(\lambda k.g(k)(x))(f). \end{aligned}$$

In the above definition, the only difficulty arises when translating the primitive recursor: To be a sound translation, $\text{rec}^b : \rho^b \rightarrow (\mathbb{N}^b \rightarrow \rho^b \rightarrow \rho^b) \rightarrow \mathbb{N}^b \rightarrow \rho^b$ has to preserve the computational rules of rec , *i.e.* rec^b should satisfy

$$\text{rec}^b(a)(f)(0^b) = a \quad \text{rec}^b(a)(f)(\text{succ } n)^b = f(n^b)(\text{rec}^b(a)(f)(n^b))$$

where $k^b : \mathbb{N}^b$ is the constant function $\lambda \alpha.k$ for any $k : \mathbb{N}$. One suitable candidate for such $\text{rec}^b(a)(f) : \mathbb{N}^b \rightarrow \rho^b$ is $\text{rec}(a)(\lambda k.f(k^b))$ but it has type $\mathbb{N} \rightarrow \rho^b$. In general, we can extend a function $g : \mathbb{N} \rightarrow \rho^b$ to $g^* : \mathbb{N}^b \rightarrow \rho^b$ such that $g^*(i^b) = g(i)$ for all $i : \mathbb{N}$, by induction on the finite type ρ . We write ke to denote the extension function ($g \mapsto g^*$) as it behaves like a *Kleisli extension* for functions $\mathbb{N} \rightarrow \rho^b$. However, in general, our b-translation does not seem to be a monad, let alone a functor.

Our first goal is to prove certain ‘equality’ between any closed term $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ and its translation f^b so that once f^b satisfies a predicate for some property such as

continuity in Section 3 then so does f . We firstly relate terms and their b-translations using the following *parametrized logical relation* that was introduced in a version of the Agda implementation of [5].

Definition 2.2. For any $\alpha : \mathbb{N} \rightarrow \mathbb{N}$, we define a logical relation $R_\rho^\alpha \subseteq \rho^b \times \rho$ by

$$\begin{aligned} f R_{\mathbb{N}}^\alpha n &::= f(\alpha) = n \\ g R_{\sigma \rightarrow \tau}^\alpha h &::= \forall x^{\sigma^b}, y^\sigma (x R_\sigma^\alpha y \rightarrow g(x) R_\tau^\alpha h(y)). \end{aligned}$$

We may omit the subscript and simply write R^α if it can be inferred from the context.

Lemma 2.3. For any term $t : \rho$ in \mathbb{T} , we have

$$t^b R^\alpha t$$

for all $\alpha : \mathbb{N} \rightarrow \mathbb{N}$, assuming $x^b R^\alpha x$ for all $x \in \text{FV}(t)$.

Proof. Let $\alpha : \mathbb{N} \rightarrow \mathbb{N}$ be given. We carry out the proof by structural induction over t . As the others are trivial, here we prove only the case $t \equiv \text{rec}$ with the following claims:

(1) The Kleisli extension ke preserves the relation R^α , i.e.

$$\text{ke}(g) R_{\mathbb{N} \rightarrow \rho}^\alpha h$$

for any $g : \mathbb{N} \rightarrow \rho^b$ and $h : \mathbb{N} \rightarrow \rho$ with $g(i) R^\alpha h(i)$ for all $i : \mathbb{N}$.

Proof. By induction on ρ . □

(2) Given $x : \rho^b$ and $y : \rho$ with $x R^\alpha y$, and $f : \mathbb{N}^b \rightarrow \rho^b \rightarrow \rho^b$ and $g : \mathbb{N} \rightarrow \rho \rightarrow \rho$ with $f R^\alpha g$,

$$\text{rec}(x)(\lambda k. f(\lambda \alpha. k))(i) R^\alpha \text{rec}(y)(g)(i)$$

for all $i : \mathbb{N}$.

Proof. By induction on i . □

By applying (1) to (2), we get a proof of $\text{rec}^b R^\alpha \text{rec}$. □

Hence we have $f^b R^\alpha f$ for any closed $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$. Unfolding it according to Definition 2.2, we can see that a term $\Omega : \mathbb{N}^b \rightarrow \mathbb{N}^b$ with $\Omega R^\alpha \alpha$ is needed in order to get the equality $f^b(\Omega)(\alpha) = f(\alpha)$. Such a term Ω can be viewed as a *generic element* [3, 4] or *generic sequence* [5], and can be easily defined by unfolding $\Omega R^\alpha \alpha$: because $\Omega R^\alpha \alpha$ is unfolded to

$$\forall f^{(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}}, n^{\mathbb{N}} (f(\alpha) = n \rightarrow \Omega(f)(\alpha) = \alpha(n))$$

by replacing n by $f\alpha$ as they are equal by assumption, we define $\Omega : \mathbb{N}^b \rightarrow \mathbb{N}^b$ by

$$\Omega(f)(\alpha) ::= \alpha(f\alpha).$$

Then the following lemma is trivial but is necessary for deriving our first result.

Lemma 2.4. For any $\alpha : \mathbb{N} \rightarrow \mathbb{N}$, we have

$$\Omega R^\alpha \alpha.$$

Our first result follows directly from Lemmas 2.3 and 2.4.

Theorem 2.5. For any closed term $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ in \mathbb{T} ,

$$f^b(\Omega)(\alpha) = f(\alpha)$$

for all $\alpha : \mathbb{N} \rightarrow \mathbb{N}$.

3. CONTINUITY OF T-DEFINABLE FUNCTIONALS

After “precooking” \mathbb{T} as above, we can now carry out the steps of the usual syntactic method. The crucial difference is that the predicate to work with is defined instead on elements of the b-translated types, so that continuity of functions $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ becomes the base case of the predicate. With a proof by induction on terms, we show that the b-translation of any term in \mathbb{T} satisfies the predicate. The case for closed terms of type $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ will bring us the desired result.

Recall that a function $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ is *continuous* if for any sequence $\alpha : \mathbb{N} \rightarrow \mathbb{N}$ there exists $m : \mathbb{N}$, called a *modulus of continuity* of f at the point α , such that any sequence $\beta : \mathbb{N} \rightarrow \mathbb{N}$ which is equal to α up to the first m positions gives the same result. In *e.g.* HA^ω , the continuity of f can be formulated as

$$\forall \alpha^{\mathbb{N} \rightarrow \mathbb{N}} \exists m^{\mathbb{N}} \forall \beta^{\mathbb{N} \rightarrow \mathbb{N}} (\alpha =_m \beta \rightarrow f(\alpha) = f(\beta))$$

where $\alpha =_m \beta$ stands for $\forall i < m \alpha_i = \beta_i$. It is obvious that continuity is preserved under pointwise equality in the sense that if $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ is continuous then so is any function that is pointwise equal to f .

We define a continuity predicate on elements of the b-translated types and show that the b-translation of any term in \mathbb{T} satisfies the predicate.

Definition 3.1. We define a unary predicate $C_\rho \subseteq \rho^b$ inductively on ρ by

$$\begin{aligned} C_{\mathbb{N}}(f) &: \equiv f \text{ is continuous} \\ C_{\sigma \rightarrow \tau}(g) &: \equiv \forall x^{\sigma^b} (C_\sigma(x) \rightarrow C_\tau(g(x))). \end{aligned}$$

Lemma 3.2. For any term $t : \rho$ in \mathbb{T} , we have

$$C_\rho(t^b)$$

assuming $C(x^b)$ for all $x \in \text{FV}(t)$.

Proof. By induction on the term t . Here we prove only the case $t \equiv \text{rec}$ with two claims.

(1) The Kleisli extension ke preserves the predicate C , *i.e.*

$$C_{\mathbb{N} \rightarrow \rho}(\text{ke}(g))$$

for any $g : \mathbb{N} \rightarrow \rho^b$ such that $C_\rho(g(i))$ for all $i : \mathbb{N}$.

Proof. By induction on ρ . (i) $\rho = \mathbb{N}$. Given $g : \mathbb{N} \rightarrow \mathbb{N}^b$ with $g(i)$ continuous for all $i : \mathbb{N}$ and $f : \mathbb{N}^b$ continuous, we need to show that $\text{ke}(g)(f)$ is continuous. Given $\alpha : \mathbb{N} \rightarrow \mathbb{N}$, let m be the modulus of $g(f\alpha)$ at α and let n be the modulus of f at α . Take $k \equiv \max(m, n)$. Given $\beta : \mathbb{N} \rightarrow \mathbb{N}$ with $\alpha =_k \beta$, we have

$$\begin{aligned} \text{ke}(g)(f)(\alpha) &= g(f\alpha)(\alpha) \\ &= g(f\alpha)(\beta) && \text{(by the continuity of } g(f\alpha)) \\ &= g(f\beta)(\beta) && \text{(} f\alpha = f\beta \text{ by the continuity of } f) \\ &= \text{ke}(g)(f)(\beta). \end{aligned}$$

(ii) $\rho = \sigma \rightarrow \tau$. Given $g : \mathbb{N} \rightarrow \sigma^b \rightarrow \tau^b$ with $C_{\sigma \rightarrow \tau}(g(i))$ for all $i : \mathbb{N}$, $f : \mathbb{N}^b$ with $C_{\mathbb{N}}(f)$ and $a : \sigma^b$ with $C_\sigma(a)$, we have to show $C_\tau(\text{ke}(g)(f)(a))$. Define $h : \mathbb{N} \rightarrow \tau^b$ by $h(i) \equiv g(i)(a)$. Then we have $\text{ke}(h)(f) = \text{ke}(g)(f)(a)$ by definition and $C_\tau(\text{ke}(h)(f))$ by the induction hypothesis. \square

(2) Given $a : \rho^b$ with $C_\rho(a)$ and $f : \mathbb{N}^b \rightarrow \rho^b \rightarrow \rho^b$ with $C_{\mathbb{N} \rightarrow \rho \rightarrow \rho}(f)$, we have

$$C_\rho(\text{rec}(a)(\lambda k.f(\lambda \alpha.k))(i))$$

for all $i : \mathbb{N}$.

Proof. By induction on i . □

By applying (1) to (2), we get a proof of $C(\text{rec}^b)$. □

Another important fact is that the generic sequence also satisfies the predicate C .

Lemma 3.3. *We have*

$$C_{\mathbb{N} \rightarrow \mathbb{N}}(\Omega).$$

Proof. Suppose a continuous $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ is given. The goal is to show that $\Omega(f)$ is also continuous. Let $\alpha : \mathbb{N} \rightarrow \mathbb{N}$ be given. By the continuity of f , we have a modulus m of f at the point α . Take $n := \max(m, f\alpha + 1)$. Given $\beta : \mathbb{N} \rightarrow \mathbb{N}$ with $\alpha =_n \beta$, we have

$$\begin{aligned} \Omega(f)(\alpha) &= \alpha(f\alpha) \\ &= \beta(f\alpha) && (\alpha =_n \beta \text{ and } f\alpha < f\alpha + 1 \leq n) \\ &= \beta(f\beta) && (f\alpha = f\beta \text{ by the continuity of } f) \\ &= \Omega(f)(\beta) \end{aligned}$$

and hence $\Omega(f)$ is continuous. □

Theorem 3.4. *Every T-definable function $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ is continuous.*

Proof. By Lemmas 3.2 and 3.3, we know $f^b(\Omega)$ is continuous. Then f is also continuous, because f and $f^b(\Omega)$ are pointwise equal by Theorem 2.5. □

4. OTHER NOTIONS OF CONTINUITY

4.1. T-definable moduli of continuity. It is also well-known that any T-definable function $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ has a T-definable *modulus of continuity* [20, Theorem 2.7.8], that is, a function $M : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ such that

$$\forall \alpha^{\mathbb{N} \rightarrow \mathbb{N}}, \beta^{\mathbb{N} \rightarrow \mathbb{N}} (\alpha =_{M(\alpha)} \beta \rightarrow f(\alpha) = f(\beta)).$$

Here we present two ways to recover this fact.

The first approach was suggested by Ulrich Kohlenbach. During a workshop¹ in Bonn, he pointed out that our Theorem 3.4 can be precisely formulated as

for each closed term $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ in \mathbb{T} , HA^ω proves that f is continuous

whose proof remains the same. From the HA^ω proof of the continuity of f , we can extract a closed term $M : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ in \mathbb{T} and a proof in HA^ω that M is a modulus of continuity of f , via the modified realizability [11, §5].

We can also “manually” extract moduli of continuity by combining the construction with the b -translation similarly to the construction of general-bar-recursion functionals in [13]. For this, we extend System \mathbb{T} with product type $\sigma \times \tau$ (and a pairing function $\langle -; - \rangle$) and

¹Workshop “Proofs and Computation”, July 2-6 2018, Hausdorff Research Institute of Mathematics, Bonn, Germany. Website: <https://www.him.uni-bonn.de/application/types-sets-constructions/workshop-proofs-and-computation/>.

projections), and then adapt Definition 2.1 to the following: The type translation ($\rho \mapsto \rho^{\mathbf{b}}$) becomes

$$\begin{aligned}\mathbb{N}^{\mathbf{b}} &::= ((\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}) \times ((\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}) \\ (\sigma \rightarrow \tau)^{\mathbf{b}} &::= \sigma^{\mathbf{b}} \rightarrow \tau^{\mathbf{b}}.\end{aligned}$$

For simplicity, we omit the trivial translation of product types and related constants. Note that each term w of type $\mathbb{N}^{\mathbf{b}}$ denotes a pair of functionals. We write

- $V_w : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ for the first component of w , and
- $M_w : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ for the second,

and hence have $w = \langle V_w; M_w \rangle$. The idea is that V_w is the *value* (i.e. the \mathbf{b} -translation) of some term while M_w is a *modulus of continuity* of V_w . Because function types are translated in the same way as in Definition 2.1, we need to change the term translation ($t \mapsto t^{\mathbf{b}}$) only for the constants of \mathbb{N} :

$$\begin{aligned}0^{\mathbf{b}} &::= \langle \lambda\alpha.0; \lambda\alpha.0 \rangle \\ \text{succ}^{\mathbf{b}} &::= \lambda x. \langle \lambda\alpha. \text{succ}(V_x(\alpha)); M_x \rangle \\ \text{rec}^{\mathbf{b}} &::= \lambda a f. \mathbf{ke}(\text{rec}(a)(\lambda k. f \langle \lambda\alpha. k; \lambda\alpha. 0 \rangle))\end{aligned}$$

where the Kleisli extension $\mathbf{ke}_\rho : (\mathbb{N} \rightarrow \rho^{\mathbf{b}}) \rightarrow \mathbb{N}^{\mathbf{b}} \rightarrow \rho^{\mathbf{b}}$ is defined by

$$\begin{aligned}\mathbf{ke}_{\mathbb{N}}(g)(f) &::= \langle \lambda\alpha. V_{g(V_f(\alpha))}(\alpha); \lambda\alpha. \max(M_{g(V_f(\alpha))}(\alpha), M_f(\alpha)) \rangle \\ \mathbf{ke}_{\sigma \rightarrow \tau}(g)(f) &::= \lambda x. \mathbf{ke}_\tau(\lambda k. g(k)(x))(f).\end{aligned}$$

Note that the maximum function $\max : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ is primitive recursive and can be defined using rec in \mathbf{T} . The “value” part of the above translation is exactly the \mathbf{b} -translation. In the “modulus” part of the Kleisli extension $\mathbf{ke}_{\mathbb{N}}(g)(f)$, there are two potential moduli of continuity at α : one is given by $g(V_f(\alpha))$ and the other by f . We of course take the greater one to be the modulus of continuity.

The generic sequence $\Omega : \mathbb{N}^{\mathbf{b}} \rightarrow \mathbb{N}^{\mathbf{b}}$ is defined by

$$\Omega(f) ::= \langle \lambda\alpha. \alpha(V_f(\alpha)); \lambda\alpha. \max(M_f(\alpha), V_f(\alpha) + 1) \rangle.$$

For the “modulus” part, we again take the greater of the two potential moduli of continuity: one is given by the input f , and the other is $V_f(\alpha) + 1$ because α is applied to $V_f(\alpha)$.

We can show the following variant of Theorem 2.5

any closed $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ in \mathbf{T} is pointwise equal to $V_{f^{\mathbf{b}}}(\Omega)$

using a parametrized logical relation $\mathbf{R}_\rho^\alpha \subseteq \rho^{\mathbf{b}} \times \rho$ that is almost the same as Definition 2.2 except that the base case is defined by

$$f \mathbf{R}_{\mathbb{N}}^\alpha n ::= V_f(\alpha) = n.$$

We also modify Definition 3.1 to get a predicate $\mathbf{C}_\rho \subseteq \rho^{\mathbf{b}}$ with the following base case

$$\mathbf{C}_{\mathbb{N}}(f) ::= M_f \text{ is a modulus of continuity of } V_f.$$

Similarly to the proof of Theorem 3.4, by showing (i) $\mathbf{C}_\rho(t^{\mathbf{b}})$ for all $t : \rho$ in \mathbf{T} and (ii) $\mathbf{C}(\Omega)$, we can conclude that for any $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ in \mathbf{T} , the term $M_{f^{\mathbf{b}}}(\Omega)$ is a modulus of continuity of f . Note that $M_{f^{\mathbf{b}}}(\Omega)$ is exactly the term which is extracted from the proof of Theorem 3.4 via modified realizability.

Theorem 4.1. *If $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ is \mathbf{T} -definable, then it has a \mathbf{T} -definable modulus of continuity.*

4.2. Uniform continuity. Because each T-definable function $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ is continuous, so is its restriction $f|_{\mathbb{2}^{\mathbb{N}}} : (\mathbb{N} \rightarrow \mathbb{2}) \rightarrow \mathbb{N}$. Using our syntactic method, we can show that these restrictions are actually uniformly continuous.

Recall that a function $f : (\mathbb{N} \rightarrow \mathbb{2}) \rightarrow \mathbb{N}$ is *uniformly continuous* if there exists $m : \mathbb{N}$, called a *modulus of uniform continuity* of f , such that any binary sequences $\alpha, \beta : \mathbb{N} \rightarrow \mathbb{2}$ that agree at the first m positions give the same result. The uniform continuity of f can be formulated in HA^ω as

$$\exists m^{\mathbb{N}} \forall \alpha^{\mathbb{N} \rightarrow \mathbb{2}}, \beta^{\mathbb{N} \rightarrow \mathbb{2}} (\alpha =_m \beta \rightarrow f(\alpha) = f(\beta)).$$

The following lemma of uniform continuity will play an important role in our proof.

Lemma 4.2. *For any $m : \mathbb{N}$ and $f : (\mathbb{N} \rightarrow \mathbb{2}) \rightarrow \mathbb{N}$, if m is a modulus of uniform continuity of f , then f has a maximum image.*

Proof. By induction on m . □

We sketch out our proof of uniform continuity only with the key steps. Firstly we define a predicate $\text{UC}_\rho \subseteq \rho^{\text{b}}$ by

$$\begin{aligned} \text{UC}_{\mathbb{N}}(f) &::= f|_{\mathbb{2}^{\mathbb{N}}} \text{ is uniformly continuous} \\ \text{UC}_{\sigma \rightarrow \tau}(g) &::= \forall x^{\sigma^{\text{b}}} (\text{UC}_\sigma(x) \rightarrow \text{UC}_\tau(g(x))). \end{aligned}$$

Then we show

- $\text{UC}_\rho(t^{\text{b}})$ for all $t : \rho$ in T, assuming $\text{UC}(x^{\text{b}})$ for all $x \in \text{FV}(t)$; and
- $\text{UC}_{\mathbb{N} \rightarrow \mathbb{N}}(\Omega)$.

Both are proved using Lemma 4.2. They together lead to the desired result.

Theorem 4.3. *If $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ is T-definable, then its restriction $f|_{\mathbb{2}^{\mathbb{N}}} : (\mathbb{N} \rightarrow \mathbb{2}) \rightarrow \mathbb{N}$ is uniformly continuous.*

Using the Dialectica interpretation and his pointwise version of strong majorization, Kohlenbach [10] obtains a more general result: for each term $f : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ in T, he constructs a term $\Phi : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ in T such that $\Phi(\gamma)$ is a modulus of uniform continuity of f on $\{\alpha^{\mathbb{N} \rightarrow \mathbb{N}} \mid \forall i^{\mathbb{N}} \alpha_i \leq \gamma_i\}$. Our syntactic method can also provide a construction of such moduli of uniform continuity (see our Agda implementation [23]).

5. GENERALIZATION

The key step of our syntactic approach to continuity is the b-translation: natural numbers are translated to functionals from the Baire type $\mathbb{N} \rightarrow \mathbb{N}$ so that continuity becomes the base case of the predicate to work with. The translation can be generalized by replacing the Baire type by arbitrary finite type X . In specific, the type translation $\rho \mapsto \rho^X$ is defined by

$$\begin{aligned} \mathbb{N}^X &::= X \rightarrow \mathbb{N} \\ (\sigma \rightarrow \tau)^X &::= \sigma^X \rightarrow \tau^X \end{aligned}$$

while the term translation remains the same as in Definition 2.1 (with the superscribed b replaced by X). The parametrized logical relation $\text{R}_\rho^\alpha \subseteq \rho^X \times \rho$ for the “ X -translation” is defined exactly the same as in Definition 2.2 except that the parameter α is of type X instead of $\mathbb{N} \rightarrow \mathbb{N}$, and similarly Lemma 2.3 for it can be shown with the same proof. The generic sequence Ω which in this case has type X^X (i.e. the X -translation of X) can be

defined by unfolding $\Omega R^\alpha \alpha$ when X is given concretely. For instance, when $X \equiv \mathbb{N}$ we define $\Omega : \mathbb{N} \rightarrow \mathbb{N}$ by $\Omega(n) := n$, as $\Omega R^n n$ is unfolded to $\Omega(n) = n$.

Therefore, our method can be generalized for proving various properties of T-definable functions of *arbitrary* finite type: Suppose the goal is to prove a certain property of T-definable functions $X \rightarrow \mathbb{N}$. As discussed above, we already have the “ X -translation” of T. Then we define a predicate $P_\rho \subseteq \rho^X$ whose base case $P_{\mathbb{N}}(f)$ expresses the targeting property of $f : X \rightarrow \mathbb{N}$. Once we show $P_\rho(t^X)$ for all $t : \rho$ in T, we achieve the goal. For instance, when taking $X \equiv \mathbb{N} \rightarrow \mathbb{N}$ (or $\mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbb{N}$), the above method is essentially Oliva and Steila’s technique of proving the closure property of System T terms under the rule of Spector’s bar recursion of type 0 (respectively 1) [13]. When taking $X \equiv \mathbb{N}$, we should be able to use the above method to prove Schwichtenberg’s theorem that every T-definable function $\mathbb{N} \rightarrow \mathbb{N}$ is eventually bounded by the slow growing hierarchy [17, Theorem 4.8] (but this is left as a future task). We have not found applications when X is higher than 1.

ACKNOWLEDGMENT

This research was supported by the Alexander von Humboldt Foundation. A part of this paper was written during the trimester “Types, Sets and Constructions” at the Hausdorff Research Institute for Mathematics (HIM), University of Bonn, May to August 2018. This visit was supported by HIM. Both this support and the hospitality of HIM are gratefully acknowledged. The author is grateful to Paulo Oliva and Martín Escardó for their motivations, and to Thierry Coquand, Peter Dybjer, Fredrik Nordvall Forsberg, Ulrich Kohlenbach, Sam Sanders, Helmut Schwichtenberg and the anonymous reviewers for various useful comments and suggestions.

REFERENCES

- [1] Agda Community. The Agda Wiki. Available at <http://wiki.portal.chalmers.se/agda/pmwiki.php>.
- [2] Michael J. Beeson. *Foundations of Constructive Mathematics*. Springer, 1985.
- [3] Thierry Coquand and Guilhem Jaber. A note on forcing and type theory. *Fundamenta Informaticae*, 100(1-4):43–52, 2010.
- [4] Thierry Coquand and Guilhem Jaber. A computational interpretation of forcing in type theory. In *Epistemology versus Ontology*, volume 27, pages 203–213. Springer Netherlands, 2012.
- [5] Martín Hötzel Escardó. Continuity of Gödel’s system T functionals via effectful forcing. *MFPS’2013. Electronic Notes in Theoretical Computer Science*, 298:119–141, 2013.
- [6] Martín Hötzel Escardó and Chuangjie Xu. A constructive manifestation of the Kleene–Kreisel continuous functionals. *Annals of Pure and Applied Logic*, 167(9):770–793, 2016. Fourth Workshop on Formal Topology (4WFTop).
- [7] Michael P. Fourman. Notions of choice sequence. In A.S. Troelstra and D. van Dalen, editors, *The L. E. J. Brouwer Centenary Symposium*, volume 110 of *Studies in Logic and the Foundations of Mathematics*, pages 91–105. Elsevier, 1982.
- [8] Michael P. Fourman. Continuous truth I, non-constructive objects. In G. Lolli, G. Longo, and A. Marcja, editors, *Logic Colloquium ’82*, volume 112 of *Studies in Logic and the Foundations of Mathematics*, pages 161–180. Elsevier, 1984.
- [9] Michael P. Fourman. Continuous truth II: reflections. In L. Libkin, U. Kohlenbach, and R. de Queiroz, editors, *Logic, Language, Information, and Computation (WoLLIC 2013)*, volume 8071 of *Lecture Notes in Computer Science*, pages 153–167. Springer Berlin Heidelberg, 2013.
- [10] Ulrich Kohlenbach. Pointwise hereditary majorization and some applications. *Archive for Mathematical Logic*, 31(4):227–241, 1992.
- [11] Ulrich Kohlenbach. *Applied Proof Theory: Proof Interpretations and Their Use in Mathematics*. Springer-Verlag Berlin Heidelberg, 2008.

- [12] John Longley. When is a functional program not a functional program? In *Proceedings of the Fourth ACM SIGPLAN International Conference on Functional Programming (ICFP '99)*, pages 1–7. ACM, 1999.
- [13] Paulo Oliva and Silvia Steila. A direct proof of Schwichtenberg’s bar recursion closure theorem. *The Journal of Symbolic Logic*, 83(1):70–83, 2018.
- [14] Thomas Powell. A functional interpretation with state. Accepted for Proceedings of Logic in Computer Science (LICS 2018), 2018.
- [15] Vincent Rahli and Mark Bickford. Validating Brouwer’s continuity principle for numbers using named exceptions. *Mathematical Structures in Computer Science*, 28(6):942–990, 2018.
- [16] Helmut Schwichtenberg. On bar recursion of types 0 and 1. *The Journal of Symbolic Logic*, 44(3):325–329, 1979.
- [17] Helmut Schwichtenberg. Proofs as programs. In P. Aczel, H. Simmons, and S. Wainer, editors, *Proof Theory: A selection of papers from the Leeds Proof Theory Programme 1990*, pages 79–114. Cambridge University Press, 1993.
- [18] Helmut Schwichtenberg and Stanley S. Wainer. *Proofs and Computations*. Cambridge University Press, 2012.
- [19] Richard Statman. Logical relations and the typed lambda calculus. *Information and Control*, 65:85–97, 1985.
- [20] Anne Sjerp Troelstra. Models and computability. In A. S. Troelstra, editor, *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*, pages 97–174. Springer Berlin Heidelberg, 1973.
- [21] Gerrit van der Hoeven and Ieke Moerdijk. Sheaf models for choice sequences. *Annals of Pure and Applied Logic*, 27(1):63–107, 1984.
- [22] Chuangjie Xu. *A continuous computational interpretation of type theories*. PhD thesis, School of Computer Science, University of Birmingham, 2015.
- [23] Chuangjie Xu. A Gentzen-style monadic translation of Gödel’s System T. Agda development available at <http://cj-xu.github.io/agda/ModTrans/index.html>, August 2019.
- [24] Chuangjie Xu. An agda implementation of Oliva & Steila’s “A direct proof of Schwichtenberg’s bar recursion closure theorem”. Available at <http://cj-xu.github.io/agda/BRCT/BRCT.html>, March 2018.
- [25] Chuangjie Xu. A syntactic approach to continuity of Gödel’s system T definable functionals. Agda development available at <http://cj-xu.github.io/agda/TCont/index.html>, September 2018.
- [26] Chuangjie Xu and Martín Hötzel 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.