Various Approaches to Computing Moduli of Continuity

Chuangjie Xu

fortiss GmbH

26 September – 1 October 2022
Proof and Computation, Fischbachau

The Continuity Principles

In Brouwerian intuitionistic mathematics,

- all functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are continuous, i.e.

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

- all functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous, i.e.

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

where B^A stands for $A \to B$, and $\alpha =_m \beta$ for $\forall(i < m).\alpha_i = \beta_i$.

Continuity of System T-Definable Functions

Theorem.

- All functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ definable in Gödel's System T are continuous.
- And their restriction to $2^{\mathbb{N}}$ are uniformly continuous.

This talk is to give a brief overview of various proofs of this fact, with a focus on their computational content, i.e., how moduli of continuity are computed.

Why am I interested in it?

- Relate different proofs via their computational content
- Generalize the methods for other purposes

But I don't know which proof is the first. And I don't know any applications.

Moduli of Continuity

A function $f : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ is continuous if

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

We call m a modulus of continuity of f at α.

A function $M : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ is called a modulus of continuity of $f : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ if $M\alpha$ is the modulus of continuity of f at α for all $\alpha : \mathbb{N}^{\mathbb{N}}$, i.e.,

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

We will explore various methods to compute moduli of continuity of functions that are definable in System T.

Continuity Proofs: Syntactic Approaches

Proving certain property by induction on T terms (without using models)

- A. S. Troelstra, editor. *Metamathematical investigation of intuitionistic arithmetic and analysis*. Springer-Verlag, Berlin, 1973.
- Ulrich Kohlenbach. Pointwise hereditary majorization and some applications. *Archive for Mathematical Logic*, 31(4):227–241, 1992.
- Thomas Powell. A functional interpretation with state. LICS'18, pp. 839–848, 2018.
- Chuangjie Xu. A Gentzen-style monadic translation of Gödel's System T. FSCD'20, pp. 30:1–30:17, 2020.

Continuity Proofs: Semantic Approaches

Operational semantics

- Thierry Coquand and Guilhem Jaber. A computational interpretation of forcing in type theory. In *Epistemology versus Ontology*, vol. 27, pp. 203–213. Springer Netherlands, 2012.

Denotational models

- Martín Escardó. Continuity of Gödel's system T functionals via effectful forcing. MFPS'13, Electronic Notes in Theoretical Computer Science, vol. 298, pp. 119–141, 2013.
- Sheaf models such as
  - Peter Johnstone. On a topological topos, *Proceedings of the London Mathematical Society*, s3-38:237–271, 1979.
  - Michael P. Fourman. Continuous truth I, non-constructive objects. Logic Colloquium '82, pp. 161–180. Elsevier, 1984.
  - Gerrit van der Hoeven and Ieke Moerdijk. Sheaf models for choice sequences. *Annals of Pure and Applied Logic*, 27(1):63–107, 1984.
  - Martín Escardó and Chuangjie Xu. A constructive manifestation of the Kleene–Kreisel continuous functionals. *Annals of Pure and Applied Logic*, 167(9):770–793, 2016.

Continuity Proofs: Computational Effects

- Andrej Bauer. Sometimes all functions are continuous. Blog. 2006.
- Vincent Rahli and Mark Bickford. A nominal exploration of intuitionism. CPP'16. 2016.
- Liron Cohen and Vincent Rahli. Realizing Continuity Using Stateful Computations. 2022.

Computing Moduli of Continuity: Evaluation Strategies

Call by name: [Escardó 2013], [Xu 2020], ...

Call by value: [Coquand&Jaber 2012], [Powell 2018], ...

Gödel's System T

We work with (the term language of) Gödel's System T in its λ-calculus form

$$\mathsf{T} \equiv \text{simply typed } \lambda\text{-calculus} + \mathsf{Nat} + \text{primitive recursor}.$$

It can be given by

$$\begin{array}{llll} \mathsf{Type} & \sigma, \tau & :\equiv & \mathsf{Nat} \mid \sigma \to \tau \\ \mathsf{Term} & t, u & :\equiv & x \mid \lambda x.t \mid tu \mid 0 \mid \mathrm{succ} \mid \mathrm{rec}_\sigma \end{array}$$

with the following typing rules:

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma} \qquad \frac{\Gamma, x : \sigma \vdash t : \tau}{\Gamma \vdash \lambda x.t : \sigma \to \tau} \qquad \frac{\Gamma \vdash t : \sigma \to \tau \qquad \Gamma \vdash u : \sigma}{\Gamma \vdash tu : \tau}$$

$$\frac{}{\Gamma \vdash 0 : \mathsf{Nat}} \qquad \frac{}{\Gamma \vdash \mathrm{succ} : \mathsf{Nat} \to \mathsf{Nat}}$$

$$\frac{}{\Gamma \vdash \mathrm{rec}_\sigma : \sigma \to (\mathsf{Nat} \to \sigma \to \sigma) \to \mathsf{Nat} \to \sigma}$$

where the context Γ is a list of distinct typed variables $x : \sigma$.

Gödel's System T: Some Conventions

We refer to only the well-typed terms $\Gamma \vdash t : \tau$.

We may omit the context and write $t : \tau$ or t^τ or just t.

We may omit the type script and write rec.

We may write

- $\lambda x_1 x_2 \cdots x_n.t$ instead of $\lambda x_1.\lambda x_2.\cdots \lambda x_n.t$;
- $f(a_1, a_2, \cdots, a_n)$ instead of $(((f a_1) a_2) \cdots) a_n$;
- $n + 1$ instead of $\mathrm{succ}(n)$;
- τ^σ instead of $\sigma \to \tau$;

Gödel's System T: Example

Using the primitive recursor, we can for instance define the function

$$\max : \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat}$$

that returns the greater argument by

$$\max := \mathrm{rec}_{\mathsf{Nat}\to\mathsf{Nat}}\,(\lambda n.n,\;\; \lambda n f.\mathrm{rec}_{\mathsf{Nat}}(n+1, \lambda m g.f m + 1))$$

One can easily verify that the usual defining equations of $\max$

$$\max(0,n) = n \quad \max(m,0) = m \quad \max(m+1,n+1) = \max(m,n)+1$$

using the computation rules of rec

$$\mathrm{rec}(a,f,0) = a \qquad \mathrm{rec}(a,f,n+1) = f(n,\mathrm{rec}(a,f,n)).$$

Gödel's System T: Standard Interpretation

System T can be modeled by any cartesian closed category with a natural numbers object.

In particular, we can interpret System T into the meta-language, where types are interpreted by

$$[\![\mathsf{Nat}]\!] := \mathbb{N}$$

$$[\![\sigma \to \tau]\!] := [\![\sigma]\!] \to [\![\tau]\!]$$

and terms $x_1 : \sigma_1, \ \cdots \ , \ x_n : \sigma_n \vdash t : \tau$ as functions $[\![\sigma_1]\!] \times \ldots \times [\![\sigma_n]\!] \to [\![\tau]\!]$ by recursion on t.

A function f is T-definable if we can fine a term t in T such $f = [\![t]\!]$.

When referring to a T-definable function, we required the term t to be given explicitly.

Computing Moduli of Continuity via Exception Handling

We extend System T with effects such as exceptions[1,2].

To compute the modulus of continuity of a pure $f : (\mathsf{Nat} \to \mathsf{Nat}) \to \mathsf{Nat}$ at an input $\alpha : \mathsf{Nat} \to \mathsf{Nat}$, we

- generate an impure input β that
  - returns αi if $i < k$
  - throws an exception otherwise

  for some parameter k, and
- try to compute $f \beta$ from $k = 0$
  - if it throws an exception, we catch it and try with $k + 1$.

At some point no exception happens. Then we know that the current value of k is a modulus of continuity.

[1] http://math.andrej.com/2006/03/27/sometimes-all-functions-are-continuous/
[2] Vincent Rahli and Mark Bickford. A nominal exploration of intuitionism. CPP'16.

Computing Moduli of Continuity via Exception Handling (cont.)

Why does this procedure of computing k terminate?

All terms of System T are total.

- Any closed term of type Nat is evaluated to some numeral $\mathrm{succ}^n(0)$ in finite steps.
- The computation of $f\alpha$ terminates, meaning that only finite parts of α can be accessed by f.
- A big enough k can be reached so that $f\beta$ also terminates.

The procedure $\alpha \mapsto k$ is not a pure program, i.e., the modulus is not a T term.

It does not seem efficient.

Continuity via Denotational Semantics

Idea: Suppose we have a model $\mathcal{M}$.

$$\begin{tikzcd} f = \llbracket t \rrbracket & \sim & \llbracket t \rrbracket_{\mathcal{M}} & \Longrightarrow & f \text{ is continuous} \\ & t \arrow[ul, mapsto] \arrow[ur, mapsto] & & & \end{tikzcd}$$

- Interpret terms t by $\llbracket t \rrbracket_{\mathcal{M}}$ in the model $\mathcal{M}$.
- Relate the two interpretations with some logical relation $\llbracket t \rrbracket \sim \llbracket t \rrbracket_{\mathcal{M}}$ by induction on t.
- Derive continuity of $\llbracket t \rrbracket$ from the proof of $\llbracket t \rrbracket \sim \llbracket t \rrbracket_{\mathcal{M}}$.

Escardó's Dialogue Trees

The type $\tilde{\mathbb{N}}$ of Dialogue trees[3] (for natural numbers) is inductively generated by

$$\eta : \mathbb{N} \to \tilde{\mathbb{N}} \qquad \beta : (\mathbb{N} \to \tilde{\mathbb{N}}) \to \mathbb{N} \to \tilde{\mathbb{N}}$$

Dialogue trees are decoded with an oracle $\alpha : \mathbb{N}^{\mathbb{N}}$ as follows:

$$\begin{aligned}
&\mathrm{dialogue} : \tilde{\mathbb{N}} \to (\mathbb{N} \to \mathbb{N}) \to \mathbb{N} \\
&\mathrm{dialogue}\ (\eta\ n)\ \alpha = n \\
&\mathrm{dialogue}\ (\beta\ \Phi\ x)\ \alpha = \mathrm{dialogue}\ (\Phi(\alpha x))\ \alpha
\end{aligned}$$

One can construct a function $\Omega : \tilde{\mathbb{N}} \to \tilde{\mathbb{N}}$, called a generic sequence, that codes any concrete sequence α, i.e.,

$$\begin{tikzcd}
\tilde{\mathbb{N}} \arrow[r, "\Omega"] \arrow[d, "\mathrm{dialogue}(-,\alpha)"'] & \tilde{\mathbb{N}} \arrow[d, "\mathrm{dialogue}(-,\alpha)"] \\
\mathbb{N} \arrow[r, "\alpha"'] & \mathbb{N}
\end{tikzcd}$$

[3] Martín Escardó. Continuity of Gödel's system T functionals via effectful forcing. MFPS'13.

Continuity via Dialogue Trees

We interpret System T using Dialogue trees

$$[\![\mathsf{Nat}]\!]_{\mathcal{D}} := \tilde{\mathbb{N}}$$
$$[\![\sigma \to \tau]\!]_{\mathcal{D}} := [\![\sigma]\!]_{\mathcal{D}} \to [\![\tau]\!]_{\mathcal{D}}$$

(The interpretation of terms is omitted.)

Lemma. For all $f : (\mathsf{Nat} \to \mathsf{Nat}) \to \mathsf{Nat}$, we have for all $\alpha : \mathbb{N}^{\mathbb{N}}$

$$[\![f]\!](\alpha) = \mathrm{dialogue}([\![f]\!]_{\mathcal{D}}(\Omega))(\alpha).$$

Lemma. For any dialogue tree d, its decodification $\mathrm{dialogue}(d) : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ is continuous.

Theorem. All T-definable function $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are continuous.

First Talk

Computing moduli of continuity via exception handling:

- Turn input α into an impure β that throws exceptions when the parameter k is not big enough to be a modulus of continuity at α.
- Totality of $f\alpha$ ensures that a big enough k can be found by testing $f\beta$.

Computing moduli of continuity via the dialogue-tree model:

- A generic sequence $\Omega : \tilde{\mathbb{N}} \to \tilde{\mathbb{N}}$ codes any sequence α
- $[\![f]\!]$ and $\mathrm{dialogue}([\![f]\!]_{\mathcal{D}}\Omega)$ are pointwise equal.
- The decodification $\mathrm{dialogue}(d)$ of any tree d is continuous.

Uniform Continuity

A function $f : \mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ is uniformly continuous if there is an $m : \mathbb{N}$, called the modulus of uniform continuity, such that

$$\forall (\alpha, \beta : \mathbf{2}^{\mathbb{N}}).\, (\alpha =_m \beta \to f\alpha = f\beta)\,.$$

Goal: Use C-spaces[4] to show that all functions $\mathbf{2}^{\mathbb{N}} \to \mathbb{N}$ that are definable in System T (extended with Booleans) are uniformly continuous.

The construction of C-spaces needs uniformly continuous endofunctions on $\mathbf{2}^{\mathbb{N}}$.

A function $f : \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ is uniformly continuous if

$$\forall (n : \mathbb{N}). \exists (m : \mathbb{N}). \forall (\alpha, \beta : \mathbf{2}^{\mathbb{N}}).\, (\alpha =_m \beta \to f\alpha =_n f\beta)\,.$$

Let C be the set of uniformly continuous maps $\mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$.

[4] Martín Escardó and Chuangjie Xu. A constructive manifestation of the Kleene–Kreisel continuous functionals. APAL, 167(9):770–793, 2016.

C-Spaces

A C-space is a set X equipped with a C-topology P, that is, a collection of maps $2^{\mathbb{N}} \to X$, called probes, satisfying the following conditions:

- All constant maps are in P.
- If $p \in P$ and $t : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ is uniformly continuous, then $p \circ t \in P$.
- For any $p_0, p_1 \in P$, the map $p : 2^{\mathbb{N}} \to X$ defined by $p(\alpha) = p_{\alpha_0}(\lambda i.\alpha_{i+1})$ is in P.

A C-continuous map from (X, P) to (Y, Q) is a map $f : X \to Y$ such that if $p \in P$ then $f \circ p \in Q$.

C-Spaces (cont.)

Theorem. The category of C-spaces is cartesian closed.

The C-space $(\mathbb{N}, P_{\mathbb{N}})$ a natural numbers object and $(\mathbf{2}, P_{\mathbf{2}})$ the coproduct of the terminal object, where P_X is the set of uniformly continuous maps into X.

The exponent $(\mathbf{2}, P_{\mathbf{2}})^{(\mathbb{N}, P_{\mathbb{N}})}$ is the internal Cantor space of C-spaces.

Lemma. The identity map on $\mathbf{2}^{\mathbb{N}}$ is a probe on $(\mathbf{2}, P_{\mathbf{2}})^{(\mathbb{N}, P_{\mathbb{N}})}$.

We denote this probe by Ω, i.e., the identity map with a continuity proof.

Yoneda Lemma. If $f : (\mathbf{2}, P_{\mathbf{2}})^{(\mathbb{N}, P_{\mathbb{N}})} \to (X, P)$ is C-continuous then $f \circ \Omega \in P$.

Note that f is (pointwise) equal to $f \circ \Omega$.

(The Yoneda lemma actually says more than above.)

Uniform Continuity via C-Spaces

We interpret System T in C-spaces by

$$\begin{aligned}
[\![\mathsf{Nat}]\!]_{\mathcal{C}} &:= (\mathbb{N}, P_{\mathbb{N}}) \\
[\![\mathsf{Bool}]\!]_{\mathcal{C}} &:= (\mathbb{2}, P_{\mathbb{2}}) \\
[\![\sigma \to \tau]\!]_{\mathcal{C}} &:= [\![\tau]\!]_{\mathcal{C}}^{[\![\sigma]\!]_{\mathcal{C}}}
\end{aligned}$$

and terms by C-continuous maps.

Lemma. For any closed term $f : (\mathsf{Nat} \to \mathsf{Bool}) \to \mathsf{Nat}$ in System T, its interpretation $[\![f]\!]_{\mathcal{C}} : (\mathbb{2}, P_{\mathbb{2}})^{(\mathbb{N}, P_{\mathbb{N}})} \to (\mathbb{N}, P_{\mathbb{N}})$ is a C-continuous map.

We have $[\![f]\!]$ pointwise equal to $[\![f]\!]_{\mathcal{C}}$, and thus also to $[\![f]\!]_{\mathcal{C}} \circ \Omega$.

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

Proof. By the Yoneda Lemma, $[\![f]\!]_{\mathcal{C}} \circ \Omega$ is a probe on $\mathbb{N}$, that is, a uniformly continuous map. Because pointwise equality preserves continuity, $[\![f]\!]$ is also uniformly continuous.

C-Spaces Can Do More!

The development of C-spaces is constructive. Thus we can extract a program to compute moduli of uniform continuity of T-definable functions.

There is a C-continuous map that interprets the fan functional

$$\mathrm{fan} : (\mathsf{Bool}^{\mathsf{Nat}} \to \mathsf{Nat}) \to \mathsf{Nat}$$

which computes the least moduli of uniform continuity.

C-spaces also form a model of Martin-Löf type theory without universe.

Continuity via Syntactic Translation

Idea:

$$
\begin{array}{ccccc}
f = [\![t]\!] & \sim & [\![t^T]\!] & \Longrightarrow & f \text{ is continuous} \\
\uparrow & & \uparrow & & \\
t & \longmapsto & t^T \in T & &
\end{array}
$$

- Translate term t into another term t^T.
- Relate the standard interpretations of the term and its translation with some logical relation $[\![t]\!] \sim [\![t^T]\!]$ by induction on t.
- Derive continuity of $[\![t]\!]$ from the proof of $[\![t]\!] \sim [\![t^T]\!]$.
  Derive a term from t^T that internalizes the modulus of continuity.

One Translation of System T

We want to translate Nat to pairs of terms of type $(\mathsf{Nat} \to \mathsf{Nat}) \to \mathsf{Nat}$, where the second term is a modulus of continuity of the first term[5].

For convenience, we extend System T with products ($\times$, pair, pr_1, pr_2).

For each finite type ρ we associate inductively a new one ρ^{b} as

$$\begin{aligned} \mathsf{Nat}^{\mathrm{b}} &:= ((\mathsf{Nat} \to \mathsf{Nat}) \to \mathsf{Nat}) \times ((\mathsf{Nat} \to \mathsf{Nat}) \to \mathsf{Nat}) \\ (\sigma \to \tau)^{\mathrm{b}} &:= \sigma^{\mathrm{b}} \to \tau^{\mathrm{b}}. \end{aligned}$$

We write $w \equiv \langle \mathrm{V}_w ; \mathrm{M}_w \rangle$ for $w : \mathsf{Nat}^{\mathrm{b}}$ and define $(t : \rho) \mapsto (t^{\mathrm{b}} : \rho^{\mathrm{b}})$ by

$$\begin{aligned} (x)^{\mathrm{b}} &:= x^{\mathrm{b}} & 0^{\mathrm{b}} &:= \langle \lambda\alpha.0 \,;\, \lambda\alpha.0 \rangle \\ (\lambda x.u)^{\mathrm{b}} &:= \lambda x^{\mathrm{b}}.u^{\mathrm{b}} & \mathrm{succ}^{\mathrm{b}} &:= \lambda x.\langle \mathrm{succ} \circ \mathrm{V}_x \,;\, \mathrm{M}_x \rangle \\ (fa)^{\mathrm{b}} &:= f^{\mathrm{b}} a^{\mathrm{b}} & \mathrm{rec}^{\mathrm{b}} &:= ??? \end{aligned}$$

[5] Chuangjie Xu. A syntactic approach to continuity of T-definable functionals. LMCS, 16(1): 22:1–22:11, 2020.

Translate Primitive Recursors

The translation $\mathrm{rec}^{\mathrm{b}} : \rho^{\mathrm{b}} \to (\mathsf{Nat}^{\mathrm{b}} \to \rho^{\mathrm{b}} \to \rho^{\mathrm{b}}) \to \mathsf{Nat}^{\mathrm{b}} \to \rho^{\mathrm{b}}$ has to preserve the computational rules of rec, i.e.,

$$\mathrm{rec}^{\mathrm{b}}(a)(f)(0^{\mathrm{b}}) = a \qquad \mathrm{rec}^{\mathrm{b}}(a)(f)(\mathrm{succ}\ n)^{\mathrm{b}} = f(n^{\mathrm{b}})(\mathrm{rec}^{\mathrm{b}}(a)(f)(n^{\mathrm{b}})).$$

A candidate for $\mathrm{rec}^{\mathrm{b}}(a)(f)$ is $\mathrm{rec}(a)(\lambda k.f(\langle \lambda\alpha.k\,; \lambda\alpha.0\rangle)) : \mathsf{Nat} \to \rho^{\mathrm{b}}$.

We can extend $g : \mathsf{Nat} \to \rho^{\mathrm{b}}$ to $g^* : \mathsf{Nat}^{\mathrm{b}} \to \rho^{\mathrm{b}}$ such that $\forall i.\ g^*(i^{\mathrm{b}}) = g(i)$ by induction on ρ – the Kleisli extension

$$\begin{aligned} \mathrm{ke}^{\mathsf{Nat}} &:= \lambda g^{\mathsf{Nat}\to\mathsf{Nat}^{\mathrm{b}}} w^{\mathsf{Nat}^{\mathrm{b}}}.\langle \lambda\alpha.\mathrm{V}_{g(\mathrm{V}_w\alpha)}\alpha\,; \lambda\alpha.\max(\mathrm{M}_{g(\mathrm{V}_w\alpha)}\alpha, \mathrm{M}_w\alpha)\rangle \\ \mathrm{ke}^{\sigma\to\tau} &:= \lambda g^{\mathsf{Nat}\to\sigma^{\mathrm{b}}\to\tau^{\mathrm{b}}} w^{\mathsf{Nat}^{\mathrm{b}}} x^{\sigma^{\mathrm{b}}}.\mathrm{ke}^{\tau}(\lambda k^{\mathsf{Nat}}.g(k,x), w). \end{aligned}$$

Hence, we define

$$\mathrm{rec}^{\mathrm{b}} := \lambda a f.\mathrm{ke}(\mathrm{rec}(a)(\lambda k.f(\langle \lambda\alpha.k\,; \lambda\alpha.0\rangle))).$$

Continuity via the Translation

We define the following parameterized logical relation $\mathrm{R}^{\alpha}_{\rho} \subseteq [\![\rho]\!] \times [\![\rho^{\mathrm{b}}]\!]$ for a given $\alpha : \mathbb{N}^{\mathbb{N}}$ by induction on ρ

$$n \; \mathrm{R}^{\alpha}_{\mathsf{Nat}} \; (f, M) \;:=\; f\alpha = n \wedge \forall(\beta : \mathbb{N}^{\mathbb{N}}).(\alpha =_{M\alpha} \beta \rightarrow f\alpha = f\beta)$$

$$g \; \mathrm{R}^{\alpha}_{\sigma\rightarrow\tau} \; h \;:=\; \forall x^{[\![\sigma]\!]} y^{[\![\sigma^{\mathrm{b}}]\!]} \, (x \; \mathrm{R}^{\alpha}_{\sigma} \; y \rightarrow gx \; \mathrm{R}^{\alpha}_{\tau} \; hy) \, .$$

Lemma. For any term $t : \rho$ in T, we have $[\![t]\!] \; \mathrm{R}^{\alpha}_{\rho} \; [\![t^{\mathrm{b}}]\!]$ for any $\alpha : \mathbb{N}^{\mathbb{N}}$.

We define a term $\Omega : \mathsf{Nat}^{\mathrm{b}} \rightarrow \mathsf{Nat}^{\mathrm{b}}$ internalizing the generic sequence by

$$\Omega := \lambda w^{\mathsf{Nat}^{\mathrm{b}}}.\langle \lambda\alpha.\alpha(\mathrm{V}_w\alpha) \, ; \lambda\alpha.\max(\mathrm{V}_w\alpha + 1, \mathrm{M}_w\alpha)\rangle.$$

Theorem. All T-definable function $\mathbb{N}^{\mathbb{N}} \rightarrow \mathbb{N}$ has a T-definable modulus of continuity.

Proof. For any $\alpha : \mathbb{N}^{\mathbb{N}}$, we have $\alpha \; \mathrm{R}^{\alpha}_{\mathsf{Nat}\rightarrow\mathsf{Nat}} \; [\![\Omega]\!]$ and thus $[\![f]\!]\alpha \; \mathrm{R}^{\alpha}_{\mathsf{Nat}} \; [\![f^{\mathrm{b}}\Omega]\!]$, i.e., $[\![\mathrm{M}_{f^{\mathrm{b}}\Omega}]\!]$ is a modulus of continuity of $[\![f]\!]$ for all term $f : (\mathsf{Nat} \rightarrow \mathsf{Nat}) \rightarrow \mathsf{Nat}$.

The Translation is Gentzen Style!

Write $J(\rho) \equiv ((\mathsf{Nat} \to \mathsf{Nat}) \to \rho) \times ((\mathsf{Nat} \to \mathsf{Nat}) \to \mathsf{Nat})$.

The translation of types of T (extended with products and sums) becomes

$$\mathsf{Nat}^{\mathrm{b}} := J(\mathsf{Nat}) \qquad (\sigma \to \tau)^{\mathrm{b}} := \sigma^{\mathrm{b}} \to \tau^{\mathrm{b}}$$
$$(\sigma + \tau)^{\mathrm{b}} := J(\sigma^{\mathrm{b}} + \tau^{\mathrm{b}}) \qquad (\sigma \times \tau)^{\mathrm{b}} := \sigma^{\mathrm{b}} \times \tau^{\mathrm{b}}$$

It's in the style of Gentzen's negative translation!

$$P^{\mathrm{G}} := \neg\neg P \qquad (\phi \to \psi)^{\mathrm{G}} := \phi^{\mathrm{G}} \to \psi^{\mathrm{G}}$$
$$(\phi \vee \psi)^{\mathrm{G}} := \neg\neg(\phi^{\mathrm{G}} \vee \psi^{\mathrm{G}}) \qquad (\phi \wedge \psi)^{\mathrm{G}} := \phi^{\mathrm{G}} \wedge \psi^{\mathrm{G}}$$
$$(\exists x.\phi)^{\mathrm{G}} := \neg\neg(\exists x.\phi^{\mathrm{G}}) \qquad (\forall x.\phi)^{\mathrm{G}} := \forall x.\phi^{\mathrm{G}}$$

The soundness theorem says $\mathrm{CL} \vdash \phi \iff \mathrm{ML} \vdash \phi^{\mathrm{G}}$.

A Generalization of Gentzen's Negative Translation

Gentzen's Translation can be generalized[6,7,8] by replacing $\neg\neg$ by a nucleus, that is, an endofunction j on formulas such that the followings are provable

$$\phi \to j\phi \qquad (\phi \to j\psi) \to j\phi \to j\psi \qquad (j\phi)[t/x] \leftrightarrow j(\phi[t/x])$$

The translation becomes

$$\begin{aligned} P_j^{\mathrm{G}} &:= jP & (\phi \to \psi)_j^{\mathrm{G}} &:= \phi_j^{\mathrm{G}} \to \psi_j^{\mathrm{G}} \\ (\phi \vee \psi)_j^{\mathrm{G}} &:= j(\phi_j^{\mathrm{G}} \vee \psi_j^{\mathrm{G}}) & (\phi \wedge \psi)_j^{\mathrm{G}} &:= \phi_j^{\mathrm{G}} \wedge \psi_j^{\mathrm{G}} \\ (\exists x.\phi)_j^{\mathrm{G}} &:= j(\exists x.\phi_j^{\mathrm{G}}) & (\forall x.\phi)_j^{\mathrm{G}} &:= \forall x.\phi_j^{\mathrm{G}} \end{aligned}$$

Working with different nuclei, we have

- if $j\phi = \neg\neg\phi$, then $\mathrm{CL} \vdash \phi \implies \mathrm{ML} \vdash \phi_j^{\mathrm{G}}$;
- if $j\phi = (\phi \to X) \to X$, then $\mathrm{CL} \vdash \phi \implies \mathrm{IL} \vdash \phi_j^{\mathrm{G}}$;
- if $j\phi = \phi \vee \bot$, then $\mathrm{IL} \vdash \phi \implies \mathrm{ML} \vdash \phi_j^{\mathrm{G}}$.

[6] Hajime Ishihara. A Note on the Gödel–Gentzen Translation. MLQ, 46(1):135–137, 2000.
[7] Martń Escardó and Paulo Oliva. The Peirce translation. APAL, 163(6):681–692, 2012.
[8] Benno van den Berg. A Kuroda-style j-translation. AML, 58(5-6):627–634, 2019.

A Gentzen-Style Translation of System T

We generalize the translation of T (without sum type for simplicity).

A nucleus $(\mathsf{JNat}, \eta, \kappa)$ consists of a type JNat and two T terms

$$\eta : \mathsf{Nat} \to \mathsf{JNat} \qquad \kappa : (\mathsf{Nat} \to \mathsf{JNat}) \to \mathsf{JNat} \to \mathsf{JNat}.$$

Given $(\mathsf{JNat}, \eta, \kappa)$, we translate types $\sigma \mapsto \sigma^{\mathrm{J}}$ by

$$\mathsf{Nat}^{\mathrm{J}} :\equiv \mathsf{JNat} \qquad (\sigma \to \tau)^{\mathrm{J}} :\equiv \sigma^{\mathrm{J}} \to \tau^{\mathrm{J}} \qquad (\sigma \times \tau)^{\mathrm{J}} :\equiv \sigma^{\mathrm{J}} \times \tau^{\mathrm{J}}$$

and terms $(t : \sigma) \mapsto (t^{\mathrm{J}} : \sigma^{\mathrm{J}})$ by

$$\begin{aligned} 0^{\mathrm{J}} &:\equiv \eta(0) & (x)^{\mathrm{J}} &:\equiv x^{\mathrm{J}} & \mathrm{pair}^{\mathrm{J}} &:\equiv \mathrm{pair} \\ \mathrm{succ}^{\mathrm{J}} &:\equiv \kappa(\eta \circ \mathrm{succ}) & (\lambda x.t)^{\mathrm{J}} &:\equiv \lambda x^{\mathrm{J}}.t^{\mathrm{J}} & \mathrm{pr}_i &:\equiv \mathrm{pr}_i \\ \mathrm{rec}^{\mathrm{J}} &:\equiv \lambda a f.\mathrm{ke}(\mathrm{rec}(a, f \circ \eta)) & (tu)^{\mathrm{J}} &:\equiv t^{\mathrm{J}} u^{\mathrm{J}} & & \end{aligned}$$

Kleisli Extension

Given $a : \rho^{\mathrm{J}}$ and $f : \mathsf{JNat} \to \rho^{J} \to \rho^{\mathrm{J}}$, define $\mathrm{rec}^{\mathrm{J}}(a, f) : \mathsf{JNat} \to \rho^{\mathrm{J}}$.

To extend $\mathrm{rec}(a, f \circ \eta) : \mathsf{Nat} \to \rho^{\mathrm{J}}$,

we cannot directly use $\kappa : (\mathsf{Nat} \to \mathsf{JNat}) \to \mathsf{JNat} \to \mathsf{JNat}$.

We define a term $\mathrm{ke}_\rho : (\mathsf{Nat} \to \rho^{\mathrm{J}}) \to \mathsf{JNat} \to \rho^{\mathrm{J}}$ of the Kleisli extension by induction on ρ as follows

$$\begin{aligned}
\mathrm{ke}_{\mathsf{Nat}}(f, a) &:\equiv \kappa(f, a) \\
\mathrm{ke}_{\sigma \to \tau}(f, a) &:\equiv \lambda x^{\sigma^{\mathrm{J}}}.\mathrm{ke}_\tau(\lambda n. f(n, x), a) \\
\mathrm{ke}_{\sigma \times \tau}(f, a) &:\equiv \langle \mathrm{ke}_\sigma(\mathrm{pr}_1 \circ f, a)\,;\mathrm{ke}_\tau(\mathrm{pr}_2 \circ f, a)\rangle.
\end{aligned}$$

and then use it to define $\mathrm{rec}^{\mathrm{J}}$.