Continuity of Gödel's system T functionals via effectful forcing

Martín Escardó
University of Birmingham, England

MFPS 2013, New Orleans
23 June 2013, 8.30am (!)

Summary

1. $\boxed{\text{What is intended to be interesting is the technique, not the well-known fact}}$

Fact.

The definable functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are continuous.

Their restrictions $2^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous.

Technique.

Work with a *generic sequence* $\mathbb{N}^{\mathbb{N}}$, analogous to the generic *Cohen real*.

But instead of forcing semantics, or sheaf toposes, or Kripke models, use

(i) a monadic interpretation, and

(ii) a generic effect.

Continuity

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

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

To know $f(\alpha)$, it is enough to know the first $m = m(\alpha)$ positions of the input α.

The number m depends on α.

There is no magic in computation (no crystal ball that predicts the infinite input).

Uniform continuity

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

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

To know $f(\alpha)$, it is enough to know the first m positions of α.

The number m is the same for all α (uniformity condition).

Any β that agrees with α at the first m positions gives the same result.

Apparent magic in computation (prediction of m for all inputs).

Summary

2. $\boxed{\text{Our proof is constructive in a very spartan sense}}$

E.g. Bar induction and the Fan Theorem are not needed.

No controversial constructive principles need to be invoked.

Summary

3. $\boxed{\text{We first developed the proof directly in intuitionistic type theory in Agda notation}}$

The paper writes the full proof in Agda from scratch, and runs experiments.

It also reports an informal, rigorous proof extracted from the formal proof.

Agda can help us. (Coq can help you more, and is equally constructive.)

With experience, it can be more flexible than pencil-and-paper scribbles.

And we can run the resulting proof, because it is constructive.

Agda proofs are programs in a literal sense.

Proving is functional programmming.

Well-known theorem

1. The definable functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are continuous.
2. Their restrictions $2^{\mathbb{N}} \to \mathbb{N}$ are uniformly continuous.

This holds for a variety of programming languages (all?).

To illustrate our technique, we consider the simplest, non-trivial one: system T.

Combinatory version of system T

1. Base type for $\mathbb{N}$, function types.
2. Constants for zero, successor and primitive recursion on $\mathbb{N}$.
3. Typed combinators $Kxy = x$ and $Sfgx = fx(gx)$.

We also have an Agda proof for the lambda-calculus version of system T.

Not reported in the paper, but available on the web.

No doubt the proof technique adapts to many extensions.

One possible proof of the (uniform) continuity theorem

$\boxed{\text{Cartesian closed model of continuous functions with natural numbers object}}$

E.g. Kleene–Kreisel functionals (1950's), sequential topological spaces, . . .

This works, but the available proofs are non-constructive.

The required number m is shown to exist, but is not exhibited.

(Twist in classical computability theory: after the existence proof, we can actually find it by unbounded inspection, in the case of uniform continuity.)

$\boxed{\text{We want a proof that directly exhibits } m \text{ (and perhaps efficiently)}}$

Other proofs (non-exhaustive list)

1. Troelstra 1970's (as far as I understand, relies on the Fan Theorem).
2. Beeson 1980's – forcing.
3. Coquand and Jaber 2010 give "A computational interpretation of forcing".

Our work is directly inspired by (3).

Main difference: they use operational methods, we use denotational methods.

(Both approaches are interesting, giving complementary perspectives.)

Our proof

Let t be a system T term denoting a function $f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$.

1. Replace the set $\mathbb{N}$ by a suitable set $\tilde{\mathbb{N}}$ non-standard natural numbers.

   Still interpret arrows as sets of all functions.

2. Get a function $\tilde{f} : (\tilde{\mathbb{N}} \to \tilde{\mathbb{N}}) \to \tilde{\mathbb{N}}$ from the term t.

3. There is a "generic sequence" $\mathrm{generic} : \tilde{\mathbb{N}} \to \tilde{\mathbb{N}}$.

4. The non-standard number $\tilde{f}(\mathrm{generic})$ gives global information about f.

   In particular, it gives (uniform) continuity information about f.

5. A logical relation between the $\mathbb{N}$ and $\tilde{\mathbb{N}}$ models explains (4).

Our proof

Consider dialogue trees in the sense of Kleene 1950's.

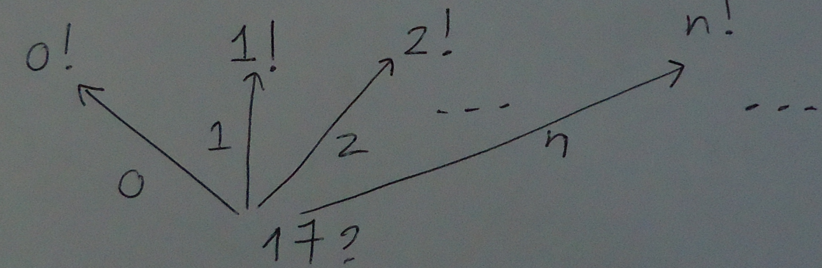
Any definable function $f : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ is computed by some well-founded dialogue.

The well-foundedness of the dialogue corresponds to the totality of f.

(Uniform) continuity information can be directly read-off from dialogue trees.

$f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ $\qquad$ $f\alpha = \alpha_{17}$

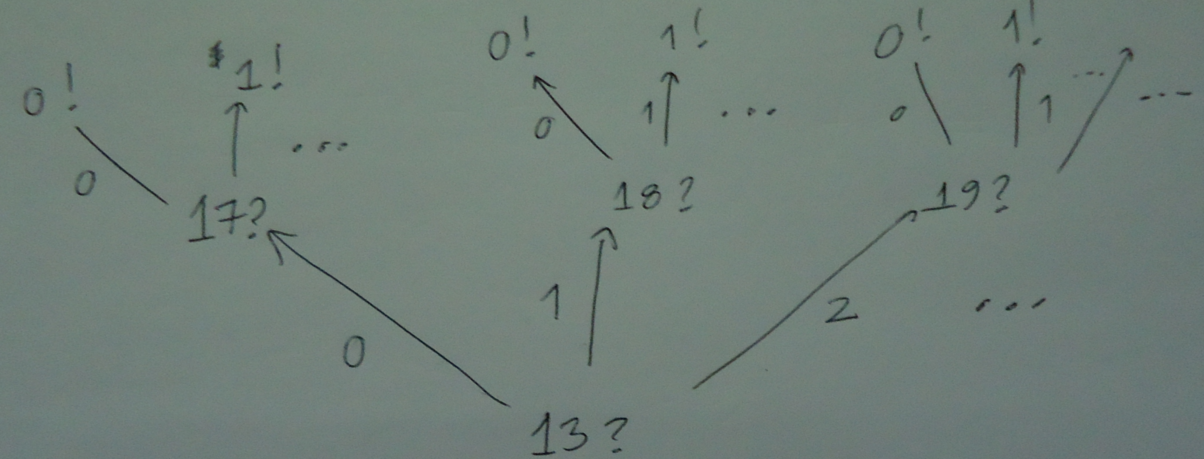
Dialogue tree of f :



$?$ = question. $\qquad$ $n?$ = what is α_n ?

$!$ = answer. $\qquad$ $m!$ = $f\alpha$ is m.

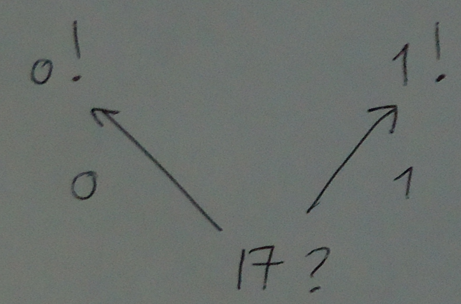
$f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ $\qquad$ $f\alpha = \alpha(17 + \alpha 13)$



<u>continuity:</u> Given α, run the dialogue.

- The modulus of continuity of f at α is the largest question plus one. (or less)
- If no questions are asked (constant function), the modulus is zero.

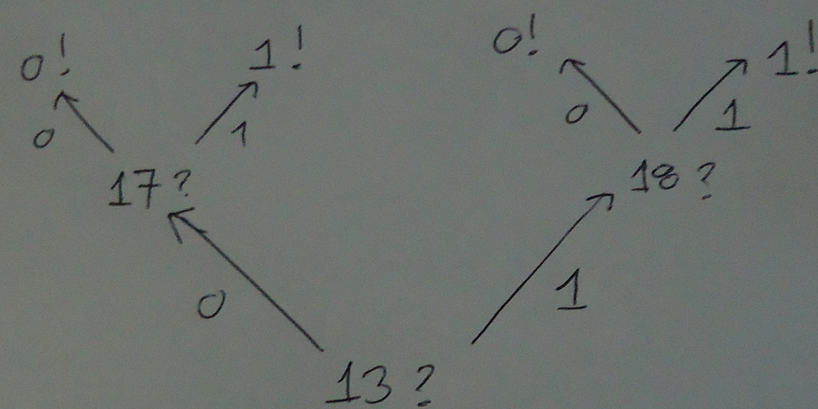
$f : (N \to 2) \to N$ $\qquad$ $f\alpha = \alpha_{17}$



Finite tree.
Because well founded & finitely branching.
(<u>No need</u> to invoke König's lemma or Fan theorem.)

Modulus of uniform continuity is 18.
Largest question plus 1

$f : (\mathbb{N} \to 2) \to \mathbb{N}$ $\qquad$ $f\alpha = \alpha(17 + \alpha 13)$



modulus of uniform continuity $= 18 + 1$

Eloquence Theorem

Every definable function $f : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ has a well-founded dialogue tree.

Hence it is continuous, and its restriction to $2^{\mathbb{N}}$ is uniformly continuous.

Continuity: we follow a finite path to find the answer.

Uniform continuity: there are finitely many finite paths to inspect.

Computing the dialogue tree of a definable function $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$

Let t be a system T term denoting a function $f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$.

1. Let $\tilde{\mathbb{N}}$ be the set of dialogue trees.
2. An auxiliary interpretation gives a function $\tilde{f} : (\tilde{\mathbb{N}} \to \tilde{\mathbb{N}}) \to \tilde{\mathbb{N}}$.
3. Ground type interpreted as $\tilde{\mathbb{N}}$, function types interpreted as sets of functions.
4. Constants interpreted as $\tilde{0}$, $\widetilde{\mathrm{Succ}}$, $\widetilde{\mathrm{Rec}}$, K, S. No tilde for K and S.
5. There is a "generic sequence" $\mathrm{generic} : \tilde{\mathbb{N}} \to \tilde{\mathbb{N}}$.
6. $\tilde{f}(\mathrm{generic})$ is a dialogue tree that computes f.

   Flavour of normalization by evaluation.
   Compute a dialogue tree by evaluating the term in a suitable model.

7. Logical relation between the $\mathbb{N}$ and $\tilde{\mathbb{N}}$ models gives the correctness of (6).

Dialogue trees for functions $(X \to Y) \to Z$

The set $D = DXYZ$ is inductively defined by the constructors

$$\begin{aligned}
\eta &: Z \to D, \\
\beta &: (Y \to D) \to X \to D.
\end{aligned}$$

We inductively define

$$\begin{array}{llllll}
\text{dialogue-function}: & D & \to & (X \to Y) & \to & Z \\
\text{dialogue-function} & (\eta\, z) & & \alpha & = & z, \\
\text{dialogue-function} & (\beta\, \phi\, x) & & \alpha & = & \text{dialogue-function } (\phi(\alpha x))\ \alpha,
\end{array}$$

We also define $\mathrm{B}\,X = D\mathbb{N}\mathbb{N}X$ and $\tilde{\mathbb{N}} = \mathrm{B}\,\mathbb{N}$, and

$$\begin{array}{llllll}
\text{decode}: & (\mathbb{N} \to \mathbb{N}) & \to & BX & \to & X, \\
\text{decode} & \alpha & & d & = & \text{dialogue-function } d\ \alpha.
\end{array}$$

The generic sequence

We have a monad B with object part defined by

$$\begin{aligned} \eta &: X \to BX, \\ \beta &: (\mathbb{N} \to BX) \to \mathbb{N} \to \mathrm{B}\,X. \end{aligned}$$

With $X = \mathbb{N}$, we have $\beta\eta : \mathbb{N} \to \mathrm{B}\,\mathbb{N}$, and so we can define

$\mathrm{generic}$ = Kleisli extension of $\beta\eta$.

Its crucial property is that

$$\begin{tikzcd} \mathrm{B}\,\mathbb{N} \arrow[r, "\mathrm{generic}"] \arrow[d, "\mathrm{decode}\,\alpha"'] & \mathrm{B}\,\mathbb{N} \arrow[d, "\mathrm{decode}\,\alpha"] \\ \mathbb{N} \arrow[r, "\alpha"'] & \mathbb{N}. \end{tikzcd}$$

Set-theoretical model of system T

$$
\begin{aligned}
[\![\iota]\!] &= \mathbb{N} \\
[\![\sigma \to \tau]\!] &= [\![\sigma]\!] \to [\![\tau]\!]
\end{aligned}
$$

Dialogue model of system T

Call-by-name monadic semantics.

$$
\begin{aligned}
\langle\!\langle \iota \rangle\!\rangle &= \mathrm{B}\,\mathbb{N} \\
\langle\!\langle \sigma \to \tau \rangle\!\rangle &= \langle\!\langle \sigma \rangle\!\rangle \to \langle\!\langle \tau \rangle\!\rangle
\end{aligned}
$$

Only the interpretation of the ground type is different from the standard set-theoretic interpretation.

Function types are still interpreted as sets of all functions.

Set-theoretic model of system T + oracle Ω

The oracle is interpreted as a free variable ranging over sequences $\alpha \in \mathbb{N}^{\mathbb{N}}$ in the model:

$$\begin{array}{rcl}
[\![\Omega]\!]\alpha & = & \alpha \\
[\![0]\!]\alpha & = & 0 \\
[\![\mathrm{succ}]\!]\alpha & = & \mathrm{succ} \\
[\![\mathrm{rec}]\!]\alpha & = & \mathrm{rec} \\
[\![\mathrm{K}]\!]\alpha & = & \mathrm{K} \\
[\![\mathrm{S}]\!]\alpha & = & \mathrm{S} \\
[\![tu]\!]\alpha & = & [\![t]\!]\alpha([\![u]\!]\alpha)
\end{array}$$

Dialogue model of system T + oracle Ω

The oracle is interpreted as the generic sequence:

$$
\begin{array}{rcl}
\langle\!\langle \Omega \rangle\!\rangle & = & \mathrm{generic} \\
\langle\!\langle 0 \rangle\!\rangle & = & \eta 0 \\
\langle\!\langle \mathrm{succ} \rangle\!\rangle & = & B\,\mathrm{succ} \\
\langle\!\langle \mathrm{rec} \rangle\!\rangle & = & \text{some kind of Kleisli extension of } \mathrm{rec} \\
\langle\!\langle \mathrm{K} \rangle\!\rangle & = & \mathrm{K} \\
\langle\!\langle \mathrm{S} \rangle\!\rangle & = & \mathrm{S} \\
\langle\!\langle t u \rangle\!\rangle & = & \langle\!\langle t \rangle\!\rangle \langle\!\langle u \rangle\!\rangle
\end{array}
$$

Calculating dialogue trees

For any T-term $t : (\iota \to \iota) \to \iota$, we have $t\Omega : \iota$ and so we can define

$$\begin{aligned} \text{dialogue-tree}\, t &= \langle\!\langle t \rangle\!\rangle\, \text{generic} \\ &= \langle\!\langle t\Omega \rangle\!\rangle \end{aligned}$$

We need to show that

$$[\![t]\!] = \text{dialogue-function}(\text{dialogue-tree}\, t).$$

Correctness

We use a logical relation $R_\sigma \subseteq (\mathbb{N}^\mathbb{N} \to [\![\sigma]\!]) \times \langle\!\langle \sigma \rangle\!\rangle$.

$$\begin{array}{rcl} R_\iota f d & \iff & f = \text{dialogue-function}\, d \\ R_{\sigma\to\tau} f f' & \iff & \forall x : \mathbb{N}^\mathbb{N} \to [\![\sigma]\!], \forall x' : \langle\!\langle \sigma \rangle\!\rangle,\ R_\sigma x x' \to R_\tau(\lambda\alpha \to f\alpha(x\alpha))(f'x') \end{array}$$

We then prove that for any type σ and any term $t : \sigma$

$$R_\sigma\ [\![t]\!]\ \langle\!\langle t \rangle\!\rangle.$$

Considering $\sigma = \iota$, we get the desired result.

Technical summary

1. Let t be a system T term denoting a function $f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$.
2. Let $\tilde{\mathbb{N}}$ be the set of dialogue trees.
3. An auxiliary interpretation gives a function $\tilde{f} : (\tilde{\mathbb{N}} \to \tilde{\mathbb{N}}) \to \tilde{\mathbb{N}}$.
4. Ground type interpreted as $\tilde{\mathbb{N}}$, function types interpreted as sets of functions.
5. Constants interpreted as $\tilde{0}$, $\widetilde{\mathrm{Succ}}$, $\widetilde{\mathrm{Rec}}$, K, S. No tilde for K and S.
6. There is a "generic sequence" $\mathrm{generic} : \tilde{\mathbb{N}} \to \tilde{\mathbb{N}}$.
7. $\tilde{f}(\mathrm{generic})$ is a dialogue tree that computes f.

   Flavour of normalization by evaluation.
   Compute a dialogue tree by evaluating the term in a suitable model.
8. Logical relation between the $\mathbb{N}$ and $\tilde{\mathbb{N}}$ models gives the correctness of (7).