Realizability: a short course
Lecture 3: Realizability and higher type computability

John Longley
Laboratory for Foundations of Computer Science
University of Edinburgh

Higher type computability

For (total or partial) functions $\mathbb{N} \to \mathbb{N}$, all sensible definitions of computability agree. But what should 'computability' mean e.g. for operations acting on operations acting on $\ldots \mathbb{N}$?

(Motivations for this question include wanting realizability interpretations for logics!)

Many choices, e.g.:

- Partial or total operations? Arguments partial or total?
- Arguments 'computable' or of some more general kind?
- How are operations 'given' to us? Oracles / infinite graphs / algorithms / programs?
- Style of computation, e.g. sequential or parallel.
- Are operations *extensional* (i.e. functions) or not?

Headline: There *are* different (good) notions of higher type computability, but not that many.

Total type structures

In this lecture, we'll restrict attention to hereditarily total functionals over $\mathbb{N}$.

Consider TPCAs over the simple types: $\sigma ::= \iota \mid \sigma \to \tau$. An N-TPCA is a TPCA A in which there are elements

$$\begin{array}{rcl} \widehat{0}, \widehat{1}, \widehat{2}, \ldots & \in & A_\iota \\ suc & \in & A_{\iota\to\iota} \\ rec_\sigma & \in & A_{\sigma\to(\iota\to\sigma\to\sigma)\to\iota\to\sigma} \quad \text{(for each } \sigma\text{)} \end{array}$$

satisfying the obvious laws.

A total type structure (TTS) is a total, extensional N-TPCA A ($(\forall x. f \bullet x = g \bullet x) \Rightarrow f = g$) in which every element of A_ι is a numeral $\widehat{n}$.

Question: What interesting TTSs of 'computable functionals' are there?

Extensional collapse

From any N-TPCA A (including untyped PCAs!), we can obtain a TTS EC(A) $\equiv A/\sim$, where

$$\widehat{n} \sim_\iota \widehat{n} \quad \text{and} \quad \text{that's all}$$
$$f \sim_{\sigma\to\tau} g \quad \text{iff} \quad \forall x, y.\ x \sim_\sigma y \Rightarrow f \bullet x \sim_\tau g \bullet y$$

Note that EC(A) is the type structure over N in PER(A). Clearly we have a canonical simulation ϵ_A : EC(A) $\longrightarrow\!\!\!\!\rhd\ A$.

[Warning: EC($-$) is highly 'non-functorial'. $A \simeq B$ does imply EC(A) $\cong$ EC(B), but e.g. $A \subseteq B$ doesn't imply EC(A) $\hookrightarrow A$.]

So we seemingly have lots of ways of building TTSs. But how many different TTSs do we get?

Total continuous functionals

We'll start by looking at TTSs of 'continuous' functionals.

Intuition: 'Computable' functionals should be automatically 'continuous': any finite amount of output info is produced 'within finite time', and so can only depend on finitely much input info. Furthermore, 'continuous' often implies 'computable relative to some oracle $\mathbb{N} \to \mathbb{N}$'.

In 1959, two definitions of a TTS were given:

- Kleene: countable functionals.
- Kreisel: continuous functionals.

These turn out to coincide (non-trivial fact). Call this TTS C.

Many other characterizations of C have since been found.

Kleene's definition via associates

Idea: Any total continuous $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ can be 'coded' by some $f : \mathbb{N} \to \mathbb{N}$, e.g.:

$$f\langle n_0, \ldots, n_{r-1}\rangle \;=\; \begin{cases} m+1 & \text{if } F(g) = m \text{ for all } g \text{ with } \forall i < r.\ g(i) = n_i \\ 0 & \text{if there's no such } m \end{cases}$$

Reversing this idea, can define an 'application' $| : \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \rightharpoonup \mathbb{N}$:

$$f \mid g \;\simeq\; f(\tilde{g}(r)) - 1 \quad \text{where} \quad r = \mu r.\, f(\tilde{g}(r)) > 0$$

A small tweak gives $\bullet : \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \rightharpoonup \mathbb{N}^{\mathbb{N}}$:

$$f \bullet g \;=\; \Lambda n.\, f\langle n, \tilde{g}(r)\rangle - 1 \quad \text{where} \quad \ldots$$

$(\mathbb{N}^{\mathbb{N}}, \bullet)$ is Kleene's second model K_2.
In effect, Kleene defined $\mathsf{C} = \mathsf{EC}(K_2)$.

Kreisel's definition via neighbourhoods (Ershov version)

Let P be the N-TPCA of Scott partial continuous functionals, defined e.g. as the type structure over $\mathbb{N}_\bot$ in the category of DCPOs.

Facts: Each $x \in \mathsf{P}_\sigma$ is a directed limits of the compact elements below x. The compact elements can be described explicitly:

- In $\mathsf{P}_\iota = \mathbb{N}_\bot$, every element is compact.
- In $\mathsf{P}_{\sigma\to\tau}$, the compact elements are certain finite joins of step functions: $(a_0 \Rightarrow b_0) \sqcup \cdots \sqcup (a_{r-1} \Rightarrow b_{r-1})$, where the a_i, b_i are compacts.

Compact elements can be coded by natural numbers. Can think of compact elements $\sqsubseteq x$ as finite pieces of information about x.

We can now define $\mathsf{C} = \mathsf{EC}(\mathsf{P})$.

An 'effective' substructure

Given an N-TPCA A and a sub-N-TPCA $A^\sharp \hookrightarrow A$, can define EC($A; A^\sharp$) to consist of all equivalence classes in EC(A) that contain an element of $A^\sharp$.

(N.B. EC($A; A^\sharp$) isn't automatically extensional!)

E.g. both K_2 and P have evident effective substructures $K_2^\sharp, \mathsf{P}^\sharp$. In fact, EC($K_2; K_2^\sharp$) and EC(P; P$^\sharp$) also coincide: C$^\sharp \hookrightarrow$ C.

$\mathbf{C}^\sharp$ is extensional by the Kleene-Kreisel density theorem.

Proving equivalences

Traditional proofs of EC(K_2) $\cong$ EC(P) are indirect, e.g. via the category of limit spaces or filter spaces (Hyland 1979). Moreover, some of these proofs lose effectivity information.

However, a simple direct (and naturally effective) proof can be given using TPCA simulations.

Idea: There's a simulation $\theta : \mathsf{P} \longrightarrow\!\!\!\triangleright K_2$: any $x \in \mathsf{P}_\sigma$ is realized by any function $\mathbb{N} \to \mathbb{N}$ that enumerates (codes for) compacts $\sqsubseteq x$. Application in P is obviously 'continuous', hence realizable in K_2.

We can show that EC(K_2) $\longrightarrow\!\!\!\triangleright K_2$, EC(P) $\longrightarrow\!\!\!\triangleright$ P $\longrightarrow\!\!\!\triangleright K_2$ are isomorphic realizably in K_2. (Routine induction on types.)

Mathematical credentials of C

Is C the canonical choice for a TTS of continuous functionals? We might worry that there's a lot of choice in how to define such a TTS, e.g. via different choices of N-TPCA.

But does *every* decent 'continuous N-TPCA' lead to C? Likewise, is the 'effective substructure' always $C^\sharp$?

Continuous N-TPCAs

Say an N-TPCA A is continuous if it's equipped with a simulation $\theta : A \longrightarrow\!\triangleright K_2$ that 'respects numerals' up to translation within K_2.

Say (A, θ) is full continuous if (within K_2) we can pass from any $f : \mathbb{N} \to \mathbb{N}$ to some θ-realizer for $f \in A_{\iota \to \iota}$.

Ubiquity theorem for C (Longley 2007): If (A, θ) is any full continuous type structure with general recursors, satisfying some mild but unsightly conditions, then

$$\mathsf{EC}(K_2) \longrightarrow\!\triangleright K_2 \qquad \mathsf{EC}(A) \longrightarrow\!\triangleright A \longrightarrow\!\triangleright K_2$$

are isomorphic realizably in K_2.

Examples: Scott, stable, strongly stable domains; many game models; Böhm tree models, ...

Some history

Cook-Berger conjecture: In EC(P) $\longrightarrow\!\!\!\!\triangleright$ P, every equivalence class contains a PCF^{Ω}-definable element.

Proved by Dag Normann (1999). So we have

$$\mathsf{C} \longrightarrow\!\!\!\!\triangleright \mathsf{PCF}^{\Omega} \longrightarrow\!\!\!\!\triangleright \mathsf{P} \longrightarrow\!\!\!\!\triangleright K_2$$

In effect, Normann gave such a simulation and showed that it is realizably isomorphic to EC(P) $\longrightarrow\!\!\!\!\triangleright$ P $\longrightarrow\!\!\!\!\triangleright K_2$. [Not true if $\longrightarrow\!\!\!\!\triangleright K_2$ deleted!]

Generalized in (Longley 2007) to an arbitrary (suitable) $A \longrightarrow\!\!\!\!\triangleright K_2$ in place of P $\longrightarrow\!\!\!\!\triangleright K_2$. (Also PCF replaced by the combinatory language of N-TPCAs with recursion: very slightly weaker.)

Normann's argument

$$\mathsf{EC(P)} \xrightarrow{\epsilon_{\mathsf{P}}}\!\!\!\triangleright\ \mathsf{P} \xrightarrow{\theta}\!\!\!\triangleright\ K_2$$

Concentrate on pure types: $0 \equiv \iota$, $k + 1 \equiv k \to \iota$.

Main lemma: Suppose the simulations in question are realizably isomorphic up to type level $k - 1$. There is a PCF program $N : 1 \to k$ with the following property:

> If $\Phi \in \mathsf{C}_k$, $\dot{\Phi} \Vdash^{\epsilon_{\mathsf{P}}} \Phi$, $\nu \Vdash^{\theta} \dot{\Phi}$ and $\dot{\nu} \in \mathsf{P}_1$ represents ν, then $[\![N]\!](\dot{\nu}) \Vdash^{\epsilon_{\mathsf{P}}} \Phi$.

In other words, if $\dot{G} \Vdash^{\epsilon_{\mathsf{P}}} G \in \mathsf{C}_{k-1}$, then $[\![N]\!](\dot{\nu})(\dot{G})$ simulates the computation of $\Phi(G)$.

Normann's argument: further details

Very crudely, N searches through ν, testing each code $c \Rightarrow q$ to see if G 'satisfies' the condition c — if so, we return q.

Problem: c will have form $\langle a_0 \Rightarrow n_0, \ldots, a_{r-1} \Rightarrow n_{r-1} \rangle$, where the a_i represent partial elements. But it seems $\dot{G}$ can only safely be applied to total elements.

Solution: Apply G to carefully chosen total extensions of the a_i. These are computed using a clever recursive invocation of N itself on 'later' parts of ν.

Tricky bit: Showing the recursion bottoms out. Here we appeal to continuity in P: at some level, the a_i will approximate total elements sufficiently well that the right thing will happen anyway.

Generalized version (JRL)

Basic proof strategy and construction of N are similar to Normann's, but the proof of bottoming-out is much more subtle: with an arbitrary continuous A in place of P, there's no overt notion of 'approximation'.

However, we can show that simply by virtue of being realizable over K_2, A inherits enough 'continuity' that something similar can be made to work.

Main point: Simulations play an essential role, both in the formulation of the general result and in its proof.

Where we've got to ...

- We've seen that for 'continuous operations on continuous data', a large class of EC constructions all lead to $\mathbf{C}$.

- Similarly for 'effective operations on continuous data': they lead to $\mathbf{C}^\sharp$.

- What about 'effective operations on effective data'? E.g. the hereditarily effective operations, HEO $\equiv$ EC(K_1).

Critical example: Fan functional versus Kleene tree

In the C world, every functional $F : (\mathbb{N} \to \mathbb{B}) \to \mathbb{B}$ has a modulus of uniform continuity m:

$$\forall g, g'. \ (\forall i < m. \ g(i) = g'(i)) \Rightarrow F(g) = F(g')$$

There's even a (PCF-definable) functional in $\mathbf{C}$ that computes a suitable m given F (the fan functional).

By contrast, in HEO there are operations $(\mathbb{N} \to \mathbb{B}) \to \mathbb{B}$ that aren't uniformly continuous at all. E.g. the Kleene tree K is a computable binary tree with arbitrarily long paths, but no *computable* infinite path. Now consider $F_K : g \mapsto \mu n. \langle g(0), \ldots, g(n-1) \rangle \notin K$.

So $\mathsf{C}^\sharp$ and HEO are incompatible: indeed, the fan functional and the Kleene tree can't coexist in an 'effective' TTS.

However, we do have HEO $\cong$ EC($\mathsf{P}^\sharp$) (generalized Kreisel-Lacombe-Shoenfield theorem).

Effective N-TPCAs

Say an N-TPCA A is effective if it's equipped with a simulation $\theta : A \longrightarrow\!\!\triangleright K_1$ that respects numerals up to effective translation.

Ubiquity theorem for HEO: Suppose (A, θ) is an effective N-TPCA with general recursion satisfying two mild technical conditions. Then

$$\mathsf{EC}(K_1) \longrightarrow\!\!\triangleright K_1 \qquad \mathsf{EC}(A) \longrightarrow\!\!\triangleright A \longrightarrow\!\!\triangleright K_1$$

are realizably isomorphic in K_1.

(Idea of proof: A inherits some sort of KLS-style continuity just by being an effective N-TPCA.)

Examples: Effective analogues of all earlier examples. Also syntactic models for prog. languages, e.g. PCF+blah / $\approx_{obs}$.

Uniform programs for total functionals

We've shown that for any suitable [continuous or effective] A and any $F \in \mathsf{EC}(A)_\sigma$, there's a term M_F in PCF (or similar) such that $[\![M_F]\!] \Vdash^{\epsilon_A} F$.

With a little more care, we can get a uniform version of this: for every $F \in \mathsf{C}_\sigma$ [resp. HEO_σ] there's a term M_F such that *for any suitable* A, $[\![M_F]\!] \Vdash^{\epsilon_A} F \in \mathsf{EC}(A)_\sigma$.

Modified extensional collapse

For any N-TPCA A, define $\mathsf{Tot}(A) \hookrightarrow A$ as follows:

$$\begin{aligned} \mathsf{Tot}(A)_\iota &= \{\widehat{n} \mid n \in \mathbb{N}\} \\ \mathsf{Tot}(A)_{\sigma\to\tau} &= \{f \in A_{\sigma\to\tau} \mid \forall x \in \mathsf{Tot}(A)_\sigma.\ f \bullet x \downarrow \in \mathsf{Tot}(A)_\tau\} \end{aligned}$$

Now define $\mathsf{MEC}(A) \equiv \mathsf{EC}(\mathsf{Tot}(A))$.

Bezem (1985) showed $\mathsf{MEC}(K_2) \cong \mathsf{EC}(K_2)$ and $\mathsf{MEC}(K_1) \cong \mathsf{EC}(K_1)$.

The above ubiquity theorems don't immediately give us $\mathsf{MEC}(A) \cong \mathsf{EC}(A)$ in general, because $\mathsf{Tot}(A)$ won't have general recursion. However, by further refining our proofs we get analogues of both ubiquity theorems for MEC.

Conclusion

A wide range of extensional collapse constructions leads to a small handful of TTSs: $\mathsf{C}, \mathsf{C}^\sharp, \mathsf{HEO}$. So these are highly canonical mathematical objects. [Rather a pity from the point of view of dreaming up realizability interpretations!]

Moreover, only the continuous/effective dichotomy seems relevant: a lot of other things you'd think might make a difference (e.g. level of intensionality, style of computation) actually don't.

This contrasts sharply with the picture for partial type structures: as we'll see, 'effective vs. continuous' plays only a minor role there, but the other factors come to the fore.

Finally, we haven't considered computability on non-continuous data at all, e.g. Kleene (S1)-(S9) computability on the full set-theoretic type structure. May be touched on in final lecture.