Topology in the theory of computation

Martín Escardó

School of Computer Science, University of Birmingham, UK

21st Summer Conference on Topology and its Applications, July 6-9, 2006 Georgia Southern University, Statesboro, GA, USA

Credits

Except when explicitly stated, the results reported here are *not* mine.

I haven't attributed all results to specific authors.

Some of them follow (easily) by combining several (difficult) results.

Computable functions are continuous

"But, hey, computer data is discrete!"

History glimpses. Brouwer (1920's), Myhill/Sheperdson (1950's), Kleene/Kreisel (1960's), Nerode (1950's), Scott (1960's), and many other mathematicians, logicians and theoretical computer scientists, then, later and recently.

Higher-type computation over the natural numbers

One wishes to compute not only functions $f : \mathbb{N} \to \mathbb{N}$, but also e.g.

$$F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N},$$

$$\phi : ((\mathbb{N} \to \mathbb{N}) \to \mathbb{N}) \to \mathbb{N},$$

$$\Phi : (((\mathbb{N} \to \mathbb{N}) \to \mathbb{N}) \to \mathbb{N}) \to \mathbb{N}.$$

I'll explain why later, giving concrete examples.

Source of continuity

Consider the computation of a functional $F\colon (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$.

1. I act as a blackbox for the input, say $\alpha \in (\mathbb{N} \to \mathbb{N})$.
2. You are the computer.
3. Your task is to produce the answer, $F(\alpha) \in \mathbb{N}$.
4. You can query me about the input as many times as you wish.
5. After finitely many questions to me, you have to answer.

A meticulous definition of higher-type computability won't be necessary for our purposes.

Mini-theorem

Let $\mathbb{N}$ be the natural numbers with the discrete topology.

Let $(\mathbb{N} \to \mathbb{N})$ be the set of functions with the product topology.

Computable functionals $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ are continuous.

Computable functionals $(\mathbb{N} \to \mathbb{N})^n \to (\mathbb{N} \to \mathbb{N})$ are continuous.

Are the converses true? To some extent. Complete answer later.

Discontinuous functions are not computable

Hence, e.g. the characteristic map of equality (diagonal)

$$\chi_\Delta \colon (\mathbb{N} \to \mathbb{N}) \times (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$$

is not computable, because it is not continuous.

No surprises for the moment.

What topologies arise at higher types?

We have two scenarios, which are related.

(There are many more scenarios, but two will be enough for this talk.)

A. Consider only terminating computations.

1. This leads to Hausdorff spaces.
2. But gives rise to computational difficulties (Turing 1936).

B. Consider also non-terminating computations.

1. This (generalizes and) simplifies the theory (Kleene 1930's).
2. But also leads to unfamiliar, non-Hausdorff spaces.

Topologically unfamiliar case first

A computation of a natural number either

1. answers a natural number and halts, or else
2. loops for ever without giving an answer (we say it “diverges”).

Divergence as a first-class citizen:

1. Non-termination represented by a point $\perp$.
2. Define $\mathcal{N} = \mathbb{N} \cup \{\perp\}$.
3. Take smallest topology for which n is isolated for $n \in \mathbb{N}$.
4. Hence the only neighbourhood of $\perp$ is the whole space.
5. $\mathbb{N}$ discrete is a subspace of $\mathcal{N}$.

Higher-type computation

Inductively define computational “types” as follows:

0. The space $\mathcal{N}$ is a type.

1. If the spaces X and Y are types, then so are

   a. $X \times Y$ with product topology,

   b. $(X \to Y)$, continuous maps with compact-open topology.

$Q \subseteq X$ compact, $V \subseteq Y$ open $\implies \{f \in (X \to Y) \mid f(Q) \subseteq V\}$ open.

Facts

1. Types are second-countable, compact, locally compact, sober spaces, but not Hausdorff.

2. The evaluation map $(X \to Y) \times X \to Y$ is continuous for all types.

3. If $f\colon X \times Y \to Z$ is continuous, so is $\bar{f}\colon X \to (Y \to Z)$.

This gives interpretation of the typed λ-calculus.

Facts

4. Types have directed-complete specialization order.

$$x \sqsubseteq y \iff x \in \{y\}^-.$$

5. Each type X has a least point $\perp_X$.

6. For each type X, any continuous $f\colon X \to X$ has a least fixed point,

$$\mathrm{fix}(f) = \bigsqcup_n f^n(\perp_X).$$

7. $\mathrm{fix}\colon (X \to X) \to X$ is a continuous map.

This gives interpretation of general recursion.

Facts

8. Types are spectral spaces (Stone duals of distributive lattices).

9. Types are densely injective topological spaces.

Most of these facts are due to Scott, early 1970's.

Original construction of types by Scott 1969

Work with directed-complete partial orders with nice properties.

This belongs to the realm of domain theory.

To pass to the formulation given here, take the Scott topology.

And use the results reported in the book *Continuous lattices and domains* by GHKLMS (2003).

But in this talk I emphasize the topological view.

Cf. Nerode 1959. Some Stone spaces in recursion theory. Duke Math.

Full version of mini-theorem (various authors)

There is a tight link between computability and continuity:

1. *The higher-type computable functionals over the natural numbers are continuous.*

   (With respect to the topologies constructed above.)

2. *Every continuous functional is computable relatively to some oracle.*

   (An oracle is a blackbox $\mathbb{N} \to \mathbb{N}$ that the algorithm may query.)

Hereditarily total continuous functionals

They lead to Hausdorff spaces.

Inductively define totality as follows:

1. A point of base type $\mathcal{N}$ is total $\iff$ it is not $\perp$.

2. A point of type $X \times Y$ is total $\iff$ its projections into X and Y are total.

3. A point $f \in (X \to Y)$ is total $\iff$ $f(x)$ is total for every total $x \in X$.

Equivalence of hereditarily total functionals

Inductively define equivalence of total elements as follows:

1. Two total points of base type $\mathcal{N}$ are equivalent $\iff$ they are equal.

2. Two total points of type $X \times Y$ are equivalent $\iff$ their projections into X and Y components are equivalent.

3. Two total functions of type $(X \to Y)$ are equivalent $\iff$ they are equivalent at total points.

Theorem (Hyland, 1970's)

For any type X, define a space X' in two steps as follows:

1. First take the subspace of total points of X.
2. Then quotient by the above equivalence relation, to get X'.

(i) X' is a compactly generated Hausdorff space.

(ii) Moreover, such total types can be obtained directly by

1. starting with discrete $\mathbb{N}$, and then
2. closing under finite products and functions spaces in CGH.

Compactly generated Hausdorff spaces

Sometimes called k-*spaces* or *Kelley spaces*.

Due to Hurewicz, written down by Kelley, popularized by Steenrod.

Introduced to fix a failure of the category of topological spaces:

Get products and function spaces that obey the exponential law:

> Continuous "homotopies" $X \times Y \to Z$ are in bijection with continuous "paths" $X \to (Y \to Z)$.

In modern terminology, they form a cartesian closed category.

But the products and function spaces are unusual

k-Product: Topological product followed by k-reflection.

k-Function space: Compact-open topology followed by k-reflection.

Let X be a Hausdorff space.

1. $F \subseteq X$ is k-*closed* if $F \cap Q$ is closed for any compact $Q \subseteq X$.
2. Closed sets are clearly k-closed.
3. DEFINITION.
   i. X is a k-*space* if the converse holds.
   ii. If not, define kX to be X with this finer topology.
   iii. kX is the k-*space reflection* of X.

Side remark

Characterization of open sets of the k-product. $W \subseteq X \times Y$ is k-open iff

1. for each $y \in Y$, the set $U_y = \{x \in X \mid (x, y) \in W\}$ is open, and
2. for each Scott open set $\mathcal{U} \subseteq O(X)$, the set $\{y \in Y \mid U_y \in \mathcal{U}\}$ is open.

Escardó (2005), based on Escardó, Lawson and Simpson (2003).

Summary of general versus total computation

General computation

1. Unusual spaces arise (non-termination is the culprit).
2. But familiar products and function spaces.

Total computation

1. Familiar spaces arise (by ruling out non-termination).
2. But unusual products and function spaces.

Unfamiliarity of function spaces in CGH

Question. Is the k-space $((\mathbb{N} \to \mathbb{N}) \to \mathbb{N})$ zero-dimensional?

1. Seems innocent. (So everybody we ask has a go.)
2. Posed to a number of topologists and logicians.
3. Even forcing has been tried, to get independence.
4. Open for about 6 years. (Partial results by Nyikos.)

Computational significance:

If so, then two competing approaches to higher-type real-number computation coincide. (Bauer–Escardó–Simpson, Normann.)

Enough to ask the complete-regular reflection to be zero-dimensional.

Higher-type computation over the reals, version II

Repetition of the above story replacing $\mathcal{N}$ with $\mathcal{R}$.

Escardó (1996), Normann (2002, 2003), De Jaeger (2002), ...

$\mathcal{R} =$ compact, non-empty intervals (proposed by Scott 1970's).

1. Singletons play the role of real numbers.
2. Other intervals play the role of $\perp$ (degrees of non-termination).
3. For $U \subseteq \mathbb{R}$ open in the Euclidean topology,

   $$\{I \in \mathcal{R} \mid U \subseteq I\}$$

   is a basic open set of $\mathcal{R}$.

$\mathbb{R}$ is homeomorphically embedded into $\mathcal{R}$.

Higher-type computation over the reals, version II

The above results go through, e.g.:

Computable functionals are continuous.

1. Riemann integration $\int_0^1 : (\mathbb{R} \to \mathbb{R}) \to \mathbb{R}$ is computable.
2. Differentiation $D : (\mathbb{R} \to \mathbb{R}) \to (\mathbb{R} \to \mathbb{R})$ is not.

Continuous functionals of arbitrary type are computable wrt oracles (Escardó 1996).

Higher-type computation over the reals, version I

Several decades older than the previous version, many authors.

Reduce to computation over $\mathbb{N}$ via encodings.

Computationally, there is no difference between $\mathbb{N}$ and $\mathbb{Q}$.

Consider subset of $(\mathbb{N} \to \mathbb{Q})$ consisting of certain Cachy sequences:

$$|q_n - q_{n+1}| < 2^{-n}.$$

The limit functional of this into $\mathbb{R}$ is continuous and quotient.

Representation dictionary for computation

$$
\begin{array}{rcl}
\mathbb{R} & | & (\mathbb{N} \to \mathbb{N}) \\
(\mathbb{R} \to \mathbb{R}) & | & ((\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})) \\
((\mathbb{R} \to \mathbb{R}) \to \mathbb{R}) & | & (((\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})) \to (\mathbb{N} \to \mathbb{N}))
\end{array}
$$

To compute on the lhs, write algorithms with types on the lhs.

Theorem (Bauer–Escardó–Simpson 2002)

1. The two approaches coincide up to type $((\mathbb{R} \to \mathbb{R}) \to \mathbb{R})$.

2. If (the completely-regular reflection of) $((\mathbb{N} \to \mathbb{N}) \to \mathbb{N})$ is zero-dimensional, then they coincide at the next type level.

3. (Normann 2005) If (the completely-regular reflection of) all types over $\mathbb{N}$ are zero-dimensional, then the two approaches coincide at all type levels.

4. (Normann 2005) The converse (with parenthetical conditions required) holds.

So, please somebody settle the zero-dimensionality question!

A perhaps surprising fact

We had, unsurprisingly:

For $X = \mathbb{N}$, the characteristic map of equality (diagonal)

$$\chi_\Delta \colon (X \to \mathbb{N}) \times (X \to \mathbb{N}) \to \mathbb{N}$$

is not computable.

However:

For $X = (\mathbb{N} \to 2)$ (Cantor space), this is computable.

How come? The problem seems even harder!

Cantor space is compact (which $\mathbb{N}$ of course is not).

Can reduce infinitely many tests to finitely many.

A related, perhaps surprising fact

Let $\mathbb{K} = (\mathbb{N} \to 2)$.

The characteristic functional of universal quantification over $\mathbb{K}$,

$$\begin{aligned} (\mathbb{K} \to 2) &\to 2 \\ p &\mapsto \begin{cases} 1 & \text{if } p(\alpha) = 1 \text{ forall } \alpha \in \mathbb{K}, \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

is computable.

This is a second example of computers checking infinitely many cases in finite time, thanks to topological considerations.

A related, unsurprising fact

Let $\mathbb{B} = (\mathbb{N} \to \mathbb{N})$ (Baire space).

The characteristic functional of universal quantification over $\mathbb{B}$,

$$\begin{aligned} (\mathbb{B} \to 2) \quad &\to \quad 2 \\ p \quad &\mapsto \quad \begin{cases} 1 & \text{if } p(\alpha) = 1 \text{ forall } \alpha \in \mathbb{K}, \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

is not computable (because it is not continuous).

Baire space is not compact.

Practice

1. Several modern programming languages include higher-types.

   E.g. Haskell, ML.

2. A number of research centres have implemented exact real-number computation in the above fashions.

3. Some uses of topology in computation:

   a. Discover new (non-)computability results.

      (Dis)prove continuity.

      Exploit compactness and other topological notions.

   b. Establish mathematical correctness of higher-type programs.

To conclude

This was a glimpse at a fraction of what is known, what is open, and what is being done regarding the role of topology in computation.

Self-advertisement:

1. M.H. Escardó. *Synthetic topology of data types and classical spaces.* ENTCS, Elsevier, volume 87, pages 21-156, November 2004.

2. M.H. Escardó and W.K. Ho. *Operational domain theory and topology of a sequential programming language.* LICS, IEEE, June 2005, pages 427-436.

3. M.H. Escardó. *Notes on compactness*, April 2005, unpublished.

(Available from my web page `http://www.cs.bham.ac.uk/~mhe/papers/`.)