Part I: Computability on uncountable structures

Arno Pauly

Swansea University

Minicourse *Computable Analysis*, Helsinki University

Coding in the classical setting

- Turing machines give us a notion of computability over finite sequences.
- We want computability over integers, graphs, matrices, logical formulas, finite groups, ...
- So we need to code the objects of interest over finite sequences.
- This is almost always utterly straight-forward.
- But what if we want to compute with objects from uncountable spaces such as real numbers?

Overview

- **Day 1** Computability on Uncountable Structures: Computability on Cantor space, represented spaces and synthetic topology
- **Day 2** Examples: The continuous functions on the unit interval; probability measures; the space of countable ordinals
- **Day 3** Non-computability: An introduction to Weihrauch degrees

Outline

Type-2 Machines

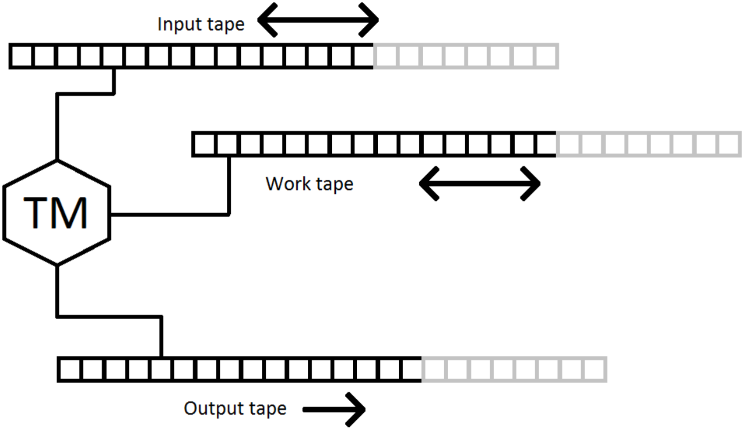

Figure: A Type-2 Machine

Alternative definition: Turing functionals

Definition

An Oracle-TM M computes $f :\subseteq \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$, if whenever M is provided $p \in \mathsf{dom}(f)$ as an oracle, it computes the total function $n \mapsto f(p)(n)$.

$\{0,1\}^{\mathbb{N}}$ carries the topology induced by the metric $d(p,q) = 2^{-\min\{n \mid p(n) \neq q(n)\}}$.

Theorem

A function $f :\subseteq \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$ is continuous iff it is computable relative to some oracle.

Represented spaces and computability

Definition

A *represented space* $\mathbf{X}$ is a pair (X, δ_X) where X is a set and $\delta_X :\subseteq \{0,1\}^{\mathbb{N}} \to X$ a surjective partial function.

Definition

$F :\subseteq \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$ is a realizer of $f :\subseteq \mathbf{X} \rightrightarrows \mathbf{Y}$, iff $\delta_Y(F(p)) \in f(\delta_X(p))$ for all $p \in \mathrm{dom}(f\delta_X)$.

$$\begin{array}{ccc} \{0,1\}^{\mathbb{N}} & \xrightarrow{F} & \{0,1\}^{\mathbb{N}} \\ \downarrow \delta_X & & \downarrow \delta_Y \\ \mathbf{X} & \xrightarrow{f} & \mathbf{Y} \end{array}$$

Definition

$f :\subseteq \mathbf{X} \rightrightarrows \mathbf{Y}$ is called computable (continuous), iff it has a computable (continuous) realizer.

A first attempt at a definition

Idea: Just use the decimal representation of reals as a coding scheme.

Alan Turing.
On computable numbers, with an application to the Entscheidungsproblem.
Proceedings of the LMS 1936.

Why that does not work

Can we multiply by 3?
Input : 0.333333333333333333333333 . . .

If we output 0.9, we could read a 4 next.

If we output 1.0, we could read a 2 next.

$\Rightarrow$: We cannot.

A first attempt at a definition

Idea: Code a real number $x \in \mathbb{R}$ by a sequence $(a_n)_{n \in \mathbb{N}}$ of (dyadic) rational numbers such that $|x - a_n| < 2^{-n}$.

Alan Turing.
On computable numbers, with an application to the Entscheidungsproblem: Corrections.
Proceedings of the LMS 1937.

This works!

Products and sums

- Let $\langle p, q\rangle \in \{0,1\}^{\mathbb{N}}$ be defined via $\langle p, q\rangle(2n) = p(n)$ and $\langle p, q\rangle(2n+1) = q(n)$.
- Given represented spaces $\mathbf{X}$ and $\mathbf{Y}$, let $\mathbf{X} \times \mathbf{Y} := (X \times Y, \delta_{X \times Y})$ be defined via $\delta_{X \times Y}(\langle p, q\rangle) = (\delta_X(p), \delta_Y(q))$.
- Moreover, let $\mathbf{X} + \mathbf{Y} := (\{0\} \times X \cup \{1\} \times Y, \delta_{X+Y})$ be defined via $\delta_{X+Y}(0p) = (0, \delta_X(p))$ and $\delta_{X+Y}(1p) = (1, \delta_Y(p))$.

Function spaces

Definition

For represented spaces **X**, **Y**, define the function space $\mathcal{C}(\mathbf{X}, \mathbf{Y})$ by $np \in \mathbb{N}^{\mathbb{N}}$ being a name of $f : X \to Y$, iff the n-th oracle machine with oracle p realizes f.

I call the elements of $\mathcal{C}(\mathbf{X}, \mathbf{Y})$ continuous functions. This is not topological continuity!

Properties of function spaces

Proposition

Let $\mathbf{X}$*,* $\mathbf{Y}$*,* $\mathbf{Z}$*,* $\mathbf{U}$ *be represented spaces. Then the following functions are computable:*

1. $\mathrm{eval} : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \times \mathbf{X} \to \mathbf{Y}$ *defined via* $\mathrm{eval}(f, x) = f(x)$.
2. $\mathrm{curry} : \mathcal{C}(\mathbf{X} \times \mathbf{Y}, \mathbf{Z}) \to \mathcal{C}(\mathbf{X}, \mathcal{C}(\mathbf{Y}, \mathbf{Z}))$ *defined via* $\mathrm{curry}(f) = x \mapsto (y \mapsto f(x, y))$.
3. $\mathrm{uncurry} : \mathcal{C}(\mathbf{X}, \mathcal{C}(\mathbf{Y}, \mathbf{Z})) \to \mathcal{C}(\mathbf{X} \times \mathbf{Y}, \mathbf{Z})$ *defined via* $\mathrm{uncurry}(f) = (x, y) \mapsto f(x)(y)$.
4. $\circ : \mathcal{C}(\mathbf{Y}, \mathbf{Z}) \times \mathcal{C}(\mathbf{X}, \mathbf{Y}) \to \mathcal{C}(\mathbf{X}, \mathbf{Z})$*, the composition of functions*
5. $\times : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \times \mathcal{C}(\mathbf{U}, \mathbf{Z}) \to \mathcal{C}(\mathbf{X} \times \mathbf{U}, \mathbf{Y} \times \mathbf{Z})$
6. $\mathrm{const} : \mathbf{Y} \to \mathcal{C}(\mathbf{X}, \mathbf{Y})$ *defined via* $\mathrm{const}(y) = (x \mapsto y)$.

Closed and open sets

Definition (Sierpiński space)

Let $\mathbb{S}$ be defined via $\delta_{\mathbb{S}}(0^{\mathbb{N}}) = 1$ and $\delta_{\mathbb{S}}(p) = 0$ for $p \neq 0^{\mathbb{N}}$.

Definition

Introduce $\mathcal{O}(\mathbf{X})$ by identifying $U \subseteq X$ with $\chi_U \in \mathcal{C}(\mathbf{X}, \mathbb{S})$.
Introduce $\mathcal{A}(\mathbf{X})$ by identifying $A \subseteq X$ with $(X \setminus A) \in \mathcal{O}(\mathbf{X})$.

Remark: The sets in $\mathcal{O}(\mathbf{X})$ happen to be the final topology on X along δ_X.

Basic operations on sets

Proposition

Let $\mathbf{X}$, $\mathbf{Y}$ be represented spaces. Then the following functions are well-defined and computable:

1. $^C : \mathcal{O}(\mathbf{X}) \to \mathcal{A}(\mathbf{X})$, $^C : \mathcal{A}(\mathbf{X}) \to \mathcal{O}(\mathbf{X})$ *mapping a set to its complement*
2. $\cup : \mathcal{O}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathcal{O}(\mathbf{X})$, $\cup : \mathcal{A}(\mathbf{X}) \times \mathcal{A}(\mathbf{X}) \to \mathcal{A}(\mathbf{X})$
3. $\cap : \mathcal{O}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathcal{O}(\mathbf{X})$, $\cap : \mathcal{A}(\mathbf{X}) \times \mathcal{A}(\mathbf{X}) \to \mathcal{A}(\mathbf{X})$
4. $\bigcup : \mathcal{C}(\mathbb{N}, \mathcal{O}(\mathbf{X})) \to \mathcal{O}(\mathbf{X})$ *mapping a sequence* $(U_n)_{n\in\mathbb{N}}$ *of open sets to their union* $\bigcup_{n\in\mathbb{N}} U_n$
5. $\bigcap : \mathcal{C}(\mathbb{N}, \mathcal{A}(\mathbf{X})) \to \mathcal{A}(\mathbf{X})$ *mapping a sequence* $(A_n)_{n\in\mathbb{N}}$ *of closed sets to their intersection* $\bigcap_{n\in\mathbb{N}} A_n$
6. $^{-1} : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \to \mathcal{C}(\mathcal{O}(\mathbf{Y}), \mathcal{O}(\mathbf{X}))$ *mapping f to f^{-1} as a set-valued function for open sets*
7. $\in : \mathbf{X} \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ *defined via* $\in (x, U) = 1$, *if* $x \in U$.
8. $\times : \mathcal{A}(\mathbf{X}) \times \mathcal{A}(\mathbf{Y}) \to \mathcal{A}(\mathbf{X} \times \mathbf{Y})$

Compactness

Definition

A represented space $\mathbf{X}$ is (computably) compact, if the map $\mathrm{isEmpty}_{\mathbf{X}} : \mathcal{A}(\mathbf{X}) \to \mathbb{S}$ defined via $\mathrm{isEmpty}_{\mathbf{X}}(\emptyset) = 1$ and $\mathrm{isEmpty}_{\mathbf{X}}(A) = 0$ otherwise is continuous (computable).

Equivalent characterizations

1. $\mathbf{X}$ is (computably) compact.
2. $\mathrm{IsFull}_{\mathbf{X}} : \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ is continuous (computable).
3. For every (computable) $A \in \mathcal{A}(\mathbf{X})$ the subspace $\mathbf{A}$ is (computably) compact.
4. $\subseteq : \mathcal{A}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ is continuous (computable).
5. $\mathrm{IsCover} : \mathcal{C}(\mathbb{N}, \mathcal{O}(\mathbf{X})) \to \mathbb{S}$ is continuous (computable).
6. $\mathrm{FiniteSubcover} :\subseteq \mathcal{C}(\mathbb{N}, \mathcal{O}(\mathbf{X})) \rightrightarrows \mathbb{N}$ is continuous (computable).
7. $\mathrm{Enough} :\subseteq \mathcal{C}(\mathbb{N}, \mathcal{A}(\mathbf{X})) \rightrightarrows \mathbb{N}$ is continuous (computable).
8. For all $\mathbf{Y}$, the map $\pi_2 : \mathcal{A}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{A}(\mathbf{Y})$ is continuous (computable).
9. For some non-empty $\mathbf{Y}$ (containing a computable point), the map $\pi_2 : \mathcal{A}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{A}(\mathbf{Y})$ is continuous (computable).

Compact sets

Definition

Define $\mathcal{K}(\mathbf{X})$ by identifying $K \subseteq X$ with $\{U \in \mathcal{O}(\mathbf{X}) \mid K \subseteq U\} \in \mathcal{O}(\mathcal{O}(\mathbf{X}))$ whenever $K = \bigcap\{U \in \mathcal{O}(\mathbf{X}) \mid K \subseteq U\}$.

Proposition

The following are computable:

1. $\subseteq : \mathcal{K}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$
2. $\cup : \mathcal{K}(\mathbf{X}) \times \mathcal{K}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$
3. $\cap : \mathcal{K}(\mathbf{X}) \times \mathcal{A}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$
4. $^{-1} :\subseteq \mathcal{C}(\mathcal{O}(\mathbf{Y}), \mathcal{O}(\mathbf{X})) \to \mathcal{C}(\mathcal{K}(\mathbf{X}), \mathcal{K}(\mathbf{Y}))$
5. $f \mapsto f : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \to \mathcal{C}(\mathcal{K}(\mathbf{X}), \mathcal{K}(\mathbf{Y}))$

Proposition

$\mathbf{X}$ *is computably compact iff* $\mathrm{id} : \mathcal{A}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$ *is well-defined and computable.*

Overt spaces

Definition
Call $\mathbf{X}$ *computably overt*, iff $\mathsf{isNotEmpty}_{\mathbf{X}} : \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ is computable.

Remark: $\mathsf{isNotEmpty}_{\mathbf{X}} : \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ is always continuous.

Overt sets

Definition

Define $\mathcal{V}(\mathbf{X})$ by identifying $A \subseteq X$ with $\{U \in \mathcal{O}(\mathbf{X}) \mid A \cap U \neq \emptyset\} \in \mathcal{O}(\mathcal{O}(\mathbf{X}))$ whenever $A = \overline{A}$.

Proposition

The following are computable:

1. $\mathrm{Intersects} : \mathcal{V}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$
2. $\cup : \mathcal{V}(\mathbf{X}) \times \mathcal{V}(\mathbf{X}) \to \mathcal{V}(\mathbf{X})$
3. $\overline{\bigcup} : \mathcal{C}(\mathbb{N}, \mathcal{V}(\mathbf{X})) \to \mathcal{V}(\mathbf{X})$
4. $\cap : \mathcal{V}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathcal{V}(\mathbf{X})$
5. $(f, A) \mapsto \overline{f(A)} : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \times \mathcal{V}(\mathbf{X}) \to \mathcal{V}(\mathbf{Y})$

Quantifying

Theorem

The following are computable:

1. $\exists : \mathcal{O}(\mathbf{X} \times \mathbf{Y}) \times \mathcal{V}(\mathbf{X}) \to \mathcal{O}(\mathbf{Y}$ *mapping* (R, A) *to* $\{y \in \mathbf{Y} \mid \exists x \in A\ (x, y) \in R\}$
2. $\forall : \mathcal{O}(\mathbf{X} \times \mathbf{Y}) \times \mathcal{K}(\mathbf{X}) \to \mathcal{O}(\mathbf{Y}$ *mapping* (R, A) *to* $\{y \in \mathbf{Y} \mid \forall x \in A\ (x, y) \in R\}$

T_2 separation

Definition

A represented space $\mathbf{X}$ is (computably) T_2, if the map $x \mapsto \{x\} : \mathbf{X} \to \mathcal{A}(\mathbf{X})$ is well-defined and continuous (computable).

Example

The cofinite space $\{\{n\} \in \mathcal{A}(\mathbb{N}) \mid n \in \mathbb{N}\}$ is **not** T_2.

T_2, characterization

1. $\mathbf{X}$ is computably T_2.
2. $\mathrm{id} : \mathcal{K}(\mathbf{X}) \to \mathcal{A}(\mathbf{X})$ is well-defined and computable.
3. $\cap : \mathcal{K}(\mathbf{X}) \times \mathcal{K}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$ is well-defined and computable, and $\mathbf{X}$ is T_0.
4. $\neq: \mathbf{X} \times \mathbf{X} \to \mathbb{S}$ defined via $\neq (x, x) = 0$ and $\neq (x, y) = 1$ otherwise is computable.
5. $\Delta_{\mathbf{X}} = \{(x, x) \mid x \in \mathbf{X}\} \in \mathcal{A}(\mathbf{X} \times \mathbf{X})$ is computable.

Observation

$\mathcal{K}(\mathbf{X})$ and $\mathcal{A}(\mathbf{X})$ are computably identical, iff $\mathbf{X}$ is a computably compact computable T_2 space.

There is much more…

Proposition

Let $\mathbf{X}$ *be (computably) compact and* $\mathbf{Y}$ *(computably)* T_2. *Then the continuous maps in* $\mathcal{C}(\mathbf{X}, \mathbf{Y})$ *are uniformly proper, i.e. the map* $(f, K) \mapsto f^{-1}(K) : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \times \mathcal{K}(\mathbf{Y}) \to \mathcal{K}(\mathbf{X})$ *is well-defined and continuous (computable).*

Standard injection

Definition

A computable canonic injection $\kappa : \mathbf{X} \to \mathcal{K}(\mathbf{X})$ is obtained via $x \mapsto (U \mapsto (x \in U))$. We denote its image by $\mathbf{X}_\kappa$.

Admissibility

Definition

Call $\mathbf{X}$ (computably) admissible, iff $\kappa : \mathbf{X} \to \mathbf{X}_\kappa$ is (computably) continuously invertible.

- Assume that $f : X \to Y$ is continuous w.r.t. the final topologies of $\delta_{\mathbf{X}}$ and $\delta_{\mathbf{Y}}$.
- Then $f^{-1} : \mathcal{O}(\mathbf{Y}) \to \mathcal{O}(\mathbf{Y})$ is well-defined and continuous.
- Hence $f : \mathcal{K}(\mathbf{X}) \to \mathcal{K}(\mathbf{Y})$ is well-defined and continuous.
- And by composition so is $(f \circ \kappa) : \mathbf{X} \to \mathbf{Y}_\kappa$.
- Now if $\mathbf{Y}$ is admissible, then $f : \mathbf{X} \to \mathbf{Y}$ is continuous.

Proposition

The map $f^{-1} \mapsto f : \mathcal{C}(\mathcal{O}(\mathbf{Y}), \mathcal{O}(\mathbf{X})) \to \mathcal{C}(\mathbf{X}, \mathbf{Y})$ *is well-defined and computable, provided that* $\mathbf{Y}$ *is computably admissible.*

Another warning

Even for admissible spaces, the product $\mathbf{X} \times \mathbf{Y}$ is not necessarily the topological product.

The admissible coreflector

Theorem

Let $\mathbf{X}$*,* $\mathbf{Y}$ *be represented spaces, and let* $\mathbf{Y}$ *be (computably) admissible. There is a (computable) continuous map* $\mathfrak{R} : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \to \mathcal{C}(\mathbf{X}_\kappa, \mathbf{Y})$ *such that* $f = \mathfrak{R}(f) \circ \kappa$ *for all* $f \in \mathcal{C}(\mathbf{X}, \mathbf{Y})$*.*

Remark: $\mathbf{X} \mapsto \mathbf{X}_\kappa$ behaves like an effective Kolmogorov quotient.

Example

$(\mathbb{R}_{10})_\kappa \cong \mathbb{R}$

How to use admissibility: An example

Assume that

1. We can recognize wrong solutions (i.e. the set of correct solutions is a computable set $A \in \mathcal{A}(\mathbf{X})$.
2. We have a candidate set $K \in \mathcal{K}(\mathbf{X})$
3. We know that there is a unique solution x in K.

Then x is computable!

Standard text book, very different approach

Klaus Weihrauch:
Computable Analysis.
Springer, 2000.

Primary source

Arno Pauly:
On the topological aspects of the theory of represented spaces.
Computability, 2016, http://arxiv.org/abs/1204.3763.

Functional programming, cartesian closed categories and topology

Martín Escardó:
Synthetic topology of datatypes and classical spaces.
ENTCS 2004.

Implementing compactness

Martín Escardó:
Seemingly impossible functional programs.
http://math.andrej.com/2007/09/28/seemingly-impossible-functional-programs/