Computability 5 (2016) 159–180
DOI 10.3233/COM-150049
IOS Press

On the topological aspects of the theory of represented spaces

Arno Pauly [*]
Computer Laboratory, University of Cambridge, Cambridge, UK
Arno.Pauly@cl.cam.ac.uk
www.cl.cam.ac.uk/~amp66/

Abstract. Represented spaces form the general setting for the study of computability derived from Turing machines. As such, they are the basic entities for endeavors such as computable analysis or computable measure theory. The theory of represented spaces is well-known to exhibit a strong topological flavour. We present an abstract and very succinct introduction to the field; drawing heavily on prior work by Escardó, Schröder, and others.

Central aspects of the theory are function spaces and various spaces of subsets derived from other represented spaces, and – closely linked to these – properties of represented spaces such as compactness, overtness and separation principles. Both the derived spaces and the properties are introduced by demanding the computability of certain mappings, and it is demonstrated that typically various interesting mappings induce the same property.

Keywords: Computable analysis, realizability, synthetic topology, admissible representation

1. Introduction

Just as numberings provide a tool to transfer computability from $\mathbb{N}$ (or $\{0,1\}^*$) to all sorts of countable structures; representations provide a means to introduce computability on structures of the cardinality of the continuum based on computability on Cantor space $\{0,1\}^{\mathbb{N}}$. This is of course essential for computable analysis [69], dealing with spaces such as $\mathbb{R}$, $\mathcal{C}(\mathbb{R}^n,\mathbb{R}^m)$, $\mathcal{C}^k(\mathbb{R}^n,\mathbb{R}^m)$ or general Hilbert spaces [8]. Computable measure theory (e.g. [19,31,45,61,68]), or computability on the set of countable ordinals [41,48,49] likewise rely on representations as foundation.

Essentially, by equipping a set with a representation, we arrive at a *represented space* – to some extent[1] the most general structure carrying a notion of computability that is derived from Turing machines.[2] Any represented space provides in a canonic way also function spaces and spaces of certain subsets. While the usefulness of these derived spaces will vary depending on the area of application, their development can be done while abstracting away from the specifics of the original represented space – in fact, such details have traditionally obfuscated the core proof ideas. Closely linked to the various subset spaces, also a few properties of spaces defined in terms of computability of certain maps are studied. Among these, special attention shall be drawn to admissibility, which has often been advocated as a criterion for well-behaved representations in computable analysis [59,69].

2. Connections to the literature

It is well-known that many representations in computable analysis can be characterized by extremal properties – the prototypic example being that the standard representation of the space of continuous functions carries exactly as much information as needed to make function evaluation computable. In fact, this condition not only characterizes

[*]The author has moved to the Département d'Informatique, Université libre de Bruxelles, Belgium.

[1]The notion of a multi-representation [58,71] goes beyond representations. A multi-representation of a set X is a right-total relation $\delta \subseteq \{0,1\}^{\mathbb{N}} \times X$, relating codes to the encoded elements. The distinction to representation is that a code can stand for more than one element here.

Several common examples of multi-representations have the additional feature that $\exists p \in \{0,1\}^{\mathbb{N}}\ \delta(p,x) \wedge \delta(p,y)$ defines an equivalence relation on X – hence, we can conceive of the multi-representation as an ordinary representation of the induced equivalence classes. Often we even have a canonization operation available; such as the saturation used in Section 5 and the closure used in Section 7 to obtain represented spaces.

A very different example (communicated to the author by Bauer) is the multirepresentation $\in\ : \{0,1\}^{\mathbb{N}} \rightrightarrows (\mathcal{P}(\{0,1\}^{\mathbb{N}}) \setminus \{\emptyset\})$. Examples of this kind, however, are beyond the scope of the paper.

[2]Algebraic computation models [6] as abstract generalizations of register machines are a completely different thing.

the representation, but also the set itself: Whenever a function is contained in a function space admitting computable function evaluation, it is computable w.r.t. some oracle – its name – hence, it is continuous. Consequently, rather than speaking merely about representations (of fixed sets) being characterized such, we should consider *represented spaces* – the combination of a set and a representation of it – as the fundamental objects of the characterization results.[3]

In many characterization results in the literature the representation is defined explicitly first, then the extremal property is proven. Often this is unnecessary though: The condition itself defines the desired represented space as a subspace of a suitable function space – which is available due to the UTM theorem. In fact, specifying details of standard representations beyond their characterizing property often complicates proofs of basic results, and can obfuscate the algorithmic ideas.

Developing the basic theory of represented spaces based on extremal characterizations amounts to an instantiation of Escardó's *synthetic topology* [26,29] with the category of represented spaces. As this category is particularly well-behaved, we obtain a very concise picture, and can prove a few more results on compactness and separation than available in the generic setting. Similarly, there are many parallels to Taylor's *abstract stone duality* [63,64], in comparison we again sacrifice generality for simplicity of presentation and strength of some results.

Most definitions and many results about them were already formulated by Schröder [57] before the advent of synthetic topology though. In this development, spaces were always required to satisfy some form of admissibility.[4] For our purposes, these restrictions are mostly immaterial, and will be dropped. Note that admissibility (w.r.t. a topology) turns out to be a feature of synthetic topology in general, demarking those spaces actually fully understandable in terms of their (internal) topology. For represented spaces, this can be seen as (slightly) generalizing [59]. The PhD thesis of Lietz [42] also contains relevant results on admissibility, in particular [42, Theorem 3.2.7]. A direct application of the framework of synthethic topology to represented spaces was performed by Collins [17,18] in 2010, several of the results presented here are already present in [18].

Effective compactness has been studied in [16] by Brattka and Weihrauch, and in [15] by Brattka and Presser. [74] by Grubba and Weihrauch considers representations of closed, open and compact sets characterized by extremal properties for the restricted case of countably based spaces. Effective topological separation was considered in [72,73] by Weihrauch.

A warning is due regarding a discrepancy in nomenclature: The computable elements of the space of closed subsets introduced here (i.e. the computable closed sets) are called co-c.e. closed sets by Weihrauch and others. Their c.e. closed sets are referred to as *overt* here (a term coined by Taylor, see Section 7), as they as a space lack the structure expected from closed sets. Subsequently, the sets called computably closed in Weihrauch's terminology are called computable closed and overt here.

The observation that the c.e. closed sets lack the closure properties of closed sets spawned various investigations into more restricted setting remedying the issue. An example of this is Ziegler's work on representations for regular closed subsets of a Euclidean space [76]. A second example, somewhat further removed from representations, is the observation that computable closed sets (i.e. co-c.e. closed) satisfying some topological criterion have to be computable overt sets (i.e. c.e. closed) as made by Miller and Iljazović [33,34,43,44].

As the attribution of the definitions and results is non-trivial due to parallel independent developments using slightly different formal frameworks, the discussion of the intellectual pedigree follows separately each section. In addition to work cited there, several observations presumably are folklore.

This paper is originally based on [46, Chapter 3], the PhD thesis of the author.

[3]Hence the change of the term of choice from Kreitz' and Weihrauch's *Theory of Representations* [38] to the present paper's *Theory of Represented Spaces*.

[4]The form of admissibility presented in this paper corresponds to admissibility w.r.t. some topology. Schröder also studied admissibility w.r.t. some variants of limit spaces.

3. The foundations

The central notion is the *represented space*, which is a pair $\mathbf{X} = (X, \delta_X)$ of a set X and a partial surjection $\delta_X :\subseteq \{0,1\}^{\mathbb{N}} \to X$.[5] A multi-valued function between represented spaces is a multi-valued function between the underlying sets. For $f :\subseteq \mathbf{X} \rightrightarrows \mathbf{Y}$ and $F :\subseteq \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$, we call F a realizer of f (notation $F \vdash f$), iff $\delta_Y(F(p)) \in f(\delta_X(p))$ for all $p \in \operatorname{dom}(f\delta_X)$. A map between represented spaces is called computable (continuous), iff it has a computable (continuous) realizer. Similarly, we call a point $x \in \mathbf{X}$ computable, iff there is some computable $p \in \mathbb{N}^{\mathbb{N}}$ with $\delta_{\mathbf{X}}(p) = x$. We write $\mathbf{X} \cong \mathbf{Y}$ if the two spaces are computably isomorphic, i.e. if there is a bijection $f : \mathbf{X} \to \mathbf{Y}$ that is computable and has a computable inverse. For our purposes, there is no need to distinguish computably isomorphic spaces.

Warning. A priori, the notion of a continuous map between represented spaces and a continuous map between topological spaces are distinct and should not be confused!

We consider two categories of represented spaces, one equipped with the computable maps, and one equipped with the continuous maps. We call the resulting structure a *category extension* (cf. [47,50]), as the former is a subcategory of the latter, and shares its structure (products, coproducts, exponentials) as we shall see next. In general, all our results have two instances, one for the computable and one for the continuous maps, with the proofs being identical. Essentially, the continuous case can be considered as the relativization of the computable one, as a function on Cantor space is continuous iff it is computable relative to some oracle.

Proposition 3.1. *For any two represented spaces* $\mathbf{X}, \mathbf{Y}$ *there are represented spaces* $\mathbf{X} \times \mathbf{Y}, \mathbf{X} + \mathbf{Y}$ *and computable functions* $\pi_1 : \mathbf{X} \times \mathbf{Y} \to \mathbf{X}$, $\pi_2 : \mathbf{X} \times \mathbf{Y} \to \mathbf{X}$, $\iota_1 : \mathbf{X} \to \mathbf{X} + \mathbf{Y}$, $\iota_2 : \mathbf{Y} \to \mathbf{X} + \mathbf{Y}$, *such that*:

1. *For any pair of continuous maps* $f_1 : \mathbf{Z} \to \mathbf{X}$, $f_2 : \mathbf{Z} \to \mathbf{Y}$ *there is a unique continuous map* $\langle f_1, f_2\rangle : \mathbf{Z} \to \mathbf{X} \times \mathbf{Y}$ *with* $f_i = \pi_i \circ \langle f_1, f_2\rangle$.
2. *If* f_1 *and* f_2 *are computable, so is* $\langle f_1, f_2\rangle$.
3. *For any pair of continuous maps* $f_1 : \mathbf{X} \to \mathbf{Z}$, $f_2 : \mathbf{Y} \to \mathbf{Z}$ *there is a unique continuous map* $(f_1 + f_2) : \mathbf{X} + \mathbf{Y} \to \mathbf{Z}$ *with* $f_i = (f_1 + f_2) \circ \iota_i$.
4. *If* f_1 *and* f_2 *are computable, so is* $f_1 + f_2$.

The space $\mathbf{X} + \mathbf{Y}$ *can be obtained via* $(X, \delta_X) + (Y, \delta_Y) = (X \uplus Y, \delta_{X+Y})$ *where* $\delta_{X+Y}(0p) = \delta_X(p)$ *and* $\delta_{X+Y}(1p) = \delta_Y(p)$. *The space* $\mathbf{X} \times \mathbf{Y}$ *can be obtained via* $(X, \delta_X) \times (Y, \delta_Y) = (X \times Y, \delta_{X\times Y})$ *where* $\delta_{X\times Y}(\langle p, q\rangle) = (\delta_X(p), \delta_Y(q))$.

Definition 3.2. Given a pair of represented spaces $\mathbf{X} = (X, \delta_X)$ and $\mathbf{Y} = (Y, \delta_Y)$ we define $\mathbf{X} \wedge \mathbf{Y} := (X \cap Y, \delta_X \wedge \delta_Y)$ where $(\delta_X \wedge \delta_Y)(\langle p, q\rangle) = x$ iff $\delta_X(p) = x \wedge \delta_Y(q) = x$.

Given two represented spaces $\mathbf{X}, \mathbf{Y}$ we obtain a third represented space $\mathcal{C}(\mathbf{X}, \mathbf{Y})$ of functions from X to Y by letting 0^n1p be a $[\delta_X \to \delta_Y]$-name for f, if the nth Turing machine equipped with the oracle p computes a realizer for f. As a consequence of the UTM theorem (in the form proven by Weihrauch [66]), $\mathcal{C}(-,-)$ is the exponential in the category of continuous maps between represented spaces, and the evaluation map is even computable.

Proposition 3.3. *Let* $\mathbf{X} = (X, \delta_X)$, $\mathbf{Y} = (Y, \delta_Y)$, $\mathbf{Z} = (Z, \delta_Z)$, $\mathbf{U} = (U, \delta_U)$ *be represented spaces. Then the following functions are computable*:

1. $\operatorname{eval} : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \times \mathbf{X} \to \mathbf{Y}$ *defined by* $\operatorname{eval}(f, x) = f(x)$.
2. $\operatorname{curry} : \mathcal{C}(\mathbf{X} \times \mathbf{Y}, \mathbf{Z}) \to \mathcal{C}(\mathbf{X}, \mathcal{C}(\mathbf{Y}, \mathbf{Z}))$ *defined by* $\operatorname{curry}(f) = x \mapsto (y \mapsto f(x, y))$.

[5] Our choice of Cantor space as domain for our representations is inconsequential, often Baire space $\mathbb{N}^{\mathbb{N}}$ is used instead. A recent observation by Cook and Kawamura is that using the set of regular functions on natural numbers as the foundation may be useful for complexity theory [36], again, this has no impact at all on the computability theory presented here.

(3) uncurry : $\mathcal{C}(\mathbf{X}, \mathcal{C}(\mathbf{Y}, \mathbf{Z})) \to \mathcal{C}(\mathbf{X} \times \mathbf{Y}, \mathbf{Z})$ *defined by* $\operatorname{uncurry}(f) = (x, y) \mapsto f(x)(y)$.
(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) const : $\mathbf{Y} \to \mathcal{C}(\mathbf{X}, \mathbf{Y})$ *defined by* $\operatorname{const}(y) = (x \mapsto y)$.

Proof. All items follow from standard arguments on Turing machines; one merely has to verify that the Type-2 semantics are unproblematic.

(1) By definition of $[\delta_X \to \delta_Y]$, if we apply the Turing machine with oracle specified in the $[\delta_X \to \delta_Y]$-name of f to a δ_X-name of x, we obtain a δ_Y-name of $f(x)$.
(2) From a Turing machine M we can compute a Turing machine M', such that M' on input p and oracle $\langle q, o\rangle$ simulates M on input $\langle p, q\rangle$ and oracle o.
(3) And vice versa.
(4) Composition of Turing machines is computable, and appropriate access to the two oracles can be ensured. As composition of realizers yields a realizer of the composition, this suffices.
(5) The execution of two Turing machines in parallel can be simulated by a single one, access to the oracles can be done accordingly. Products of realizers are realizers of products.
(6) This is done via a Turing machine which ignores its input and copies the oracle tape to the output tape. □

Corollary 3.4. *Let* $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ *be represented spaces. For any* $x \in X$, *the map* $\operatorname{partial}_x : \mathcal{C}(\mathbf{X} \times \mathbf{Y}, \mathbf{Z}) \to \mathcal{C}(\mathbf{Y}, \mathbf{Z})$ *defined by* $\operatorname{partial}_x(f) = (y \mapsto f(x, y))$ *is continuous. If* x *is computable, then so is* $\operatorname{partial}_x$.

Remarks. The earliest use of *represented spaces* as entities in their own right seems to be due to Brattka in 1996 [7], although the treatment there does not go beyond the introduction of computability before the setting is restricted to (countably based) topological spaces. The notion of a *representation* (with full generality[6]) precedes this by far, going back to Weihrauch [66] and Kreitz and Weihrauch [38] in 1985. The category of representation (of represented spaces) was explicitly mentioned by Bauer in 2002 [3].

That the study of computability and the study of continuity go hand in hand when it comes to the theory of representations (of represented spaces) has been noted since its beginnings in [38], and takes a very visible role e.g. in Weihrauch's books [67,69]. That this yields two categories with the same objects and the same structure was observed by Bauer (in a slightly different setting) in 1998 [1]. The observation that continuity is relativized computability, which is fundamental for this duality, seems to be a folk result[7] appearing first in the 1950's.

The operations $+$, $\times$, $\wedge$ on representations are all defined already in [38], their properties as given in Proposition 3.1 appear e.g. in [69]. The exponential is introduced in [66] (although not named as such). The category of represented spaces is identified as Cartesian closed in [3].

4. Open and closed sets

In the following, we will want to make use of two special represented spaces, $\mathbb{N} = (\mathbb{N}, \delta_{\mathbb{N}})$ and $\mathbb{S} = (\{\bot, \top\}, \delta_{\mathbb{S}})$. The representation are given by $\delta_{\mathbb{N}}(0^n 1 0^{\mathbb{N}}) = n$, $\delta_{\mathbb{S}}(0^{\mathbb{N}}) = \bot$ and $\delta_{\mathbb{S}}(p) = \top$ for $p \neq 0^{\mathbb{N}}$. It is straightforward to verify that the computability notion for the represented space $\mathbb{N}$ coincides with classical computability over the natural numbers.

The Sierpiński space $\mathbb{S}$ in turn allows us to formalize semi-decidability. The computable functions $f : \mathbb{N} \to \mathbb{S}$ are exactly those where $f^{-1}(\{\top\})$ is recursively enumerable (and thus $f^{-1}(\{\bot\})$ co-recursively enumerable). In general, for any represented space $\mathbf{X}$ we obtain two spaces of subsets of $\mathbf{X}$; the space of open sets $\mathcal{O}(\mathbf{X})$ by identifying $f \in \mathcal{C}(\mathbf{X}, \mathbb{S})$ with $f^{-1}(\{\top\})$, and the space of closed sets $\mathcal{A}(\mathbf{X})$ by identifying $f \in \mathcal{C}(\mathbf{X}, \mathbb{S})$ with $f^{-1}(\{\bot\})$. The properties of the spaces of open and closed sets follow from a few particular computable functions on Sierpiński space $\mathbb{S}$ and the function space properties in Proposition 3.3.

[6]In a more restrictive sense, Weihrauch and Schäfer used the term *representation* before [75].

[7]The difficulty in attributing the result to any specific person is also mentioned in [35].

Proposition 4.1. *The functions* $\wedge, \vee : \mathbb{S} \times \mathbb{S} \to \mathbb{S}$ *and* $\bigvee : \mathcal{C}(\mathbb{N}, \mathbb{S}) \to \mathbb{S}$ *are computable.*

Here we have $\wedge, \vee : \mathbb{S} \times \mathbb{S} \to \mathbb{S}$ defined by $\wedge(\top, \top) = \top$, $\wedge(x, y) = \bot$ for $(x, y) \neq (\top, \top)$, $\vee(\bot, \bot) = \bot$, $\vee(x, y) = \top$ for $(x, y) \neq (\bot, \bot)$. Moreover, $\bigvee$ is defined by $\bigvee((x_n)_{n\in\mathbb{N}}) = \top$ iff $\exists n_\top \in \mathbb{N}$ s.t. $x_{n_\top} = \top$, and $\bigvee((x_n)_{n\in\mathbb{N}}) = \bot$ otherwise.

Proof of Proposition 4.1. A machine realizing $\wedge$, $\vee$ simply writes 0s until it reads a non-zero value from one (for $\vee$) or both (for $\wedge$) input components, then it continues with 1s. In order to solve $\bigvee$, all (x_n) are investigated simultaneously, while 0s are written. If a 1 ever occurs anywhere in the input, the program starts writing 1s. □

We point out that neither $\neg : \mathbb{S} \to \mathbb{S}$ or $\bigwedge : \mathcal{C}(\mathbb{N}, \mathbb{S}) \to \mathbb{S}$ are computable.

Proposition 4.2. *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 by* $\in(x, U) = \top$, *if* $x \in U$.
(8) $\times : \mathcal{O}(\mathbf{X}) \times \mathcal{O}(\mathbf{Y}) \to \mathcal{O}(\mathbf{X} \times \mathbf{Y})$, $\times : \mathcal{A}(\mathbf{X}) \times \mathcal{A}(\mathbf{Y}) \to \mathcal{A}(\mathbf{X} \times \mathbf{Y})$.
(9) $\mathrm{Cut} : \mathbf{Y} \times \mathcal{O}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{O}(\mathbf{X})$ *mapping* (y, U) *to* $\{x \mid (x, y) \in U\}$.
(10) $\Pi : \mathcal{C}(\mathbb{N}, \mathcal{A}(\mathbf{X})) \to \mathcal{A}(\mathcal{C}(\mathbb{N}, \mathbf{X}))$, *where* $\Pi((A_n)_{n\in\mathbb{N}}) = \prod_{n\in\mathbb{N}} A_n$.

Proof. (1) By definition both functions are realized by $\mathrm{id}_{\{0,1\}^\mathbb{N}}$.
(2) Taking into consideration that $\mathcal{O}(\mathbf{X}), \mathcal{A}(\mathbf{X})$ may be considered as subspaces of the function space $\mathcal{C}(\mathbf{X}, \mathbb{S})$, we may realize $\cup : \mathcal{O}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathcal{O}(\mathbf{X})$ by composition (Proposition 3.3(4, 6), together with the diagonal) of $\vee$ and $\times$ from Proposition 3.3(5), likewise the composition of $\wedge$ and $\times$ realizes $\cup : \mathcal{A}(\mathbf{X}) \times \mathcal{A}(\mathbf{X}) \to \mathcal{A}(\mathbf{X})$.
(3) This follows from (1) and (2) using de Morgan's law.
(4) Composition of $\bigvee$ with the input yields the output, and is computable due to Proposition 3.3(4).
(5) This follows from (1) and (4) using de Morgan's law.
(6) Again, this is a special case of composition, which is computable due to Proposition 3.3(4).
(7) Here we have a special case of eval, which is computable due to Proposition 3.3(1).
(8) This follows from composing the computable function $\times : \mathcal{C}(\mathbf{X}, \mathbb{S}) \times \mathcal{C}(\mathbf{Y}, \mathbb{S}) \to \mathcal{C}(\mathbf{X} \times \mathbf{Y}, \mathbb{S} \times \mathbb{S})$ from Proposition 3.3(6) together with $\wedge : \mathbb{S} \times \mathbb{S} \to \mathbb{S}$ and $\vee : \mathbb{S} \times \mathbb{S} \to \mathbb{S}$ respectively.
(9) This follows from Proposition 3.3(1, 2) via $x \in \mathrm{Cut}(y, U)$ iff $(x, y) \in U$.
(10) Essentially, this is composition with $\bigvee : \mathcal{C}(\mathbb{N}, \mathbb{S}) \to \mathbb{S}$ from Proposition 4.1. □

A represented space $\mathbf{X}$ canonically induces a topological space by equipping the set X by the quotient topology $\mathcal{T}_\mathbf{X}$ of $\delta_X :\subseteq \{0, 1\}^\mathbb{N} \to X$. One can verify that $\mathcal{T}_\mathbf{X}$ is the underlying set of $\mathcal{O}(\mathbf{X})$; i.e. that the open sets of a represented space are the open sets in the induced topological space: Essentially, the open subset of a subspace of $\{0, 1\}^\mathbb{N}$ that witnesses that a set U is in $\mathcal{T}_\mathbf{X}$ can be translated into a continuous realizer of $\chi_U : \mathbf{X} \to \mathbb{S}$. Proposition 4.2(6) implies that a continuous map between represented spaces is also continuous as a map between the induced topological spaces. However, the converse is generally false.

Remarks. The introduction of Sierpiński space to the study of representations, and subsequent use to derive representations of the open and the closed subsets, is due to Schröder in 2002 [57] (cf. [15]). However, the usefulness of such definitions to obtain concise proofs has not been fully appreciated in this setting (e.g. Weihrauch and Grubba are not using it in [74] from 2009). Statements equivalent to the items of Proposition 4.2 have been proven in various restricted settings before.

Using the combination of the function space construction and the presence of Sierpiński space to obtain the space of open (closed) sets together with the standard operations on them is the fundamental idea of synthetic topology (Escardó 2004 [26]).

Collins has stated the results of Proposition 4.2(1–6, 9) as part of [18, Theorem 3.23, Theorem 3.24 and Proposition 3.26].

The space $\mathcal{A}(\mathbf{X})$ has been studied as the upper Fell topology on the hyperspace of closed sets in topology. This topology was introduced by Fell in 1962 [30]. Note that the characterization of compact sets in Section 5 is crucial for this coincidence, which has been observed before by Brattka and Presser 2003 [15]. A general source for hyperspace topologies is [5].

5. Compactness

Compactness is occasionally described as a generalization of finiteness, in that compactness means that it can be verified that a property applies to all elements of a space – so in some sense, compact spaces admit a form of exhaustive search,[8] despite potentially having uncountable cardinality. Compactness both plays a rôle as a property of spaces, as well as inducing a represented space of subsets of a given space. We provide various equivalent characterizations of the former, and list various useful computable operations on the latter.

Definition 5.1. A represented space $\mathbf{X}$ is (computably) compact, if the map $\text{IsEmpty}_{\mathbf{X}} : \mathcal{A}(\mathbf{X}) \to \mathbb{S}$ defined by $\text{IsEmpty}_{\mathbf{X}}(\emptyset) = \top$ and $\text{IsEmpty}_{\mathbf{X}}(A) = \bot$ otherwise is continuous (computable).

Proposition 5.2. *The following properties are equivalent for a represented space* $\mathbf{X}$:

(1) $\mathbf{X}$ *is* (*computably*) *compact.*
(2) $\text{IsFull}_{\mathbf{X}} : \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ *defined by* $\text{IsFull}_{\mathbf{X}}(X) = \top$ *and* $\text{IsFull}_{\mathbf{X}}(U) = \bot$ *otherwise is continuous* (*computable*).
(3) *For every* (*computable*[9]) $A \in \mathcal{A}(\mathbf{X})$ *the subspace* $\mathbf{A}$ *is* (*computably*) *compact.*
(4) $\subseteq : \mathcal{A}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ *defined by* $\subseteq(A, U) = \top$, *iff* $A \subseteq U$ *is continuous* (*computable*).
(5) $\text{IsCover} : \mathcal{C}(\mathbb{N}, \mathcal{O}(\mathbf{X})) \to \mathbb{S}$ *defined by* $\text{IsCover}((U_n)_{n\in\mathbb{N}}) = \top$, *iff* $\bigcup_{n\in\mathbb{N}} U_n = X$ *is continuous* (*computable*).
(6) $\text{FiniteSubcover} :\subseteq \mathcal{C}(\mathbb{N}, \mathcal{O}(\mathbf{X})) \rightrightarrows \mathbb{N}$ *with* $\text{dom}(\text{FiniteSubcover}) = \{(U_n)_{n\in\mathbb{N}} \mid \exists N \ \bigcup_{n\leqslant N} U_n = X\}$ *and* $N \in \text{FiniteSubcover}((U_n)_{n\in\mathbb{N}})$ *iff* $\bigcup_{n\leqslant N} U_n = X$ *is continuous* (*computable*).
(7) $\text{Enough} :\subseteq \mathcal{C}(\mathbb{N}, \mathcal{A}(\mathbf{X})) \rightrightarrows \mathbb{N}$ *with* $(A_i)_{i\in\mathbb{N}} \in \text{dom}(\text{Enough})$ *iff* $\exists N \ \bigcap_{i\leqslant N} A_i = \emptyset$, *and* $N \in \text{Enough}((A_i)_{i\in\mathbb{N}})$ *iff* $\bigcap_{i\leqslant N} A_i = \emptyset$ *is continuous* (*computable*).
(8) *For every represented spaces* $\mathbf{Y}$, *the map* $\pi_2 : \mathcal{A}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{A}(\mathbf{Y})$ *defined by* $\pi_2(A) = \{y \in \mathbf{Y} \mid \exists x \in \mathbf{X} \ (x, y) \in A\}$ *is well-defined and continuous* (*computable*).
(9) *For some non-empty represented space* $\mathbf{Y}$ (*containing a computable point*), *the map* $\pi_2 : \mathcal{A}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{A}(\mathbf{Y})$ *is well-defined and continuous* (*computable*).

Proof. (1) $\Leftrightarrow$ (2) $\text{IsEmpty}_{\mathbf{X}}$ and $\text{IsFull}_{\mathbf{X}}$ have exactly the same realizers.

(1) $\Leftrightarrow$ (3) X is always a computable element of $\mathcal{A}(\mathbf{X})$, as it is realized by the constant function $p \mapsto 0^{\mathbb{N}}$. This provides the implication (1) $\Leftrightarrow$ (3). For the other direction, note $\text{IsEmpty}_{\mathbf{A}}(B) = \text{IsEmpty}_{\mathbf{X}}(A \cap B)$, so continuity (computability) of $\text{IsEmpty}_{\mathbf{A}}$ follows from Proposition 4.2(2) together with Corollary 3.4 and Proposition 3.3(2).

(1) $\Leftrightarrow$ (4) $\emptyset \in \mathcal{O}(\mathbf{X})$ is computable by Proposition 3.3(6), so if $\subseteq$ is computable, so is $A \mapsto \subseteq(A, \emptyset)$ by Corollary 3.4. For the other direction, observe $A \subseteq B \Leftrightarrow A \cap B^C = \emptyset$, so we may use the combination of Proposition 4.2(1, 3), Corollary 3.4 and Proposition 3.3(2) to obtain computability of $\subseteq$ here.

[8]For the idea of exhaustive search of infinite sets, see [27].

[9]We remind the reader that we call $A \in \mathcal{A}(\mathbf{X})$ computable, if it is a computable name. This generalizes Weihrauch's notion of a co-c.e. closed set, not that of a computable closed set!

(2) ⇔ (5) By Proposition 3.3(6) we can compute $(U)_{n\in\mathbb{N}}$ from U, and clearly $\text{IsFull}(U) = \text{IsCover}((U)_{n\in\mathbb{N}})$. On the other hand, by Proposition 4.2(4), we can compute $\bigcup_{n\in\mathbb{N}} U_n$ from $(U_n)_{n\in\mathbb{N}}$, and $\text{IsCover}((U)_{n\in\mathbb{N}}) = \text{IsFull}(\bigcup_{n\in\mathbb{N}} U_n)$.

(5) ⇒ (6) Assume that the Turing machine M computes IsCover (potentially with access to some oracle). In order to solve FiniteSubcover, we simulate M on the input for FiniteSubcover (which is of a suitable type). As we know that the input sequence *does* cover X, we also know that M has to write a 1 eventually. When M writes the first 1, it has only read some finite prefix of the input. In particular, we may assume that M has no information at all about the U_n with $n > N$ for some $N \in \mathbb{N}$. Moreover, we can find such an N effectively from observing the simulation of M.
Now N constitutes a valid answer to $\text{FiniteSubcover}((U_n)_{n\in\mathbb{N}})$. To see this, assume the contrary. Then $\bigcup_{n=0}^{N} U_n \neq X$. Now consider the sequence $(U'_n)_{n\in\mathbb{N}}$ with $U'_n = U_n$ for $n \leqslant N$ and $U'_n = \emptyset$ otherwise. On some name for $(U'_n)_{n\in\mathbb{N}}$, M will eventually print a 1 – as it cannot distinguish $(U'_n)_{n\in\mathbb{N}}$ from $(U_n)_{n\in\mathbb{N}}$ before that. But as $(U'_n)_{n\in\mathbb{N}}$ does not constitute a cover of X, this contradicts the initial assumption.

(6) ⇒ (2) Assume that the Turing machine M computes FiniteSubcover (potentially with access to some oracle). In order to solve $\text{IsFull}(U)$, we simulate M on input $(U)_{n\in\mathbb{N}}$, which we can obtain by Proposition 3.3(6). Beside the simulation, we repeatedly write 0s on the output tape. If M ever produces some $N \in \mathbb{N}$ as output, we write a 1. This actually solves $\text{IsFull}(U)$ correctly.
If $U = X$, then $(U)_{n\in\mathbb{N}}$ is a valid input for FiniteSubcover, so eventually some $N \in \mathbb{N}$ will be produced, causing the final result to be $\top \in \mathbb{S}$. Now assume that M produces some $N \in \mathbb{N}$ on input $(U)_{n\in\mathbb{N}}$ for $U \neq X$. At the time where N has been written, M has only received information about some finite prefix of its input. In particular, there is some $K > N$ s.t. M exhibits the same behaviour when faced with the input sequence $(U'_n)_{n\in\mathbb{N}}$ with $U'_n = U$ for $n \leqslant K$ and $U'_n = X$ otherwise. As $(U'_n)_{n\in\mathbb{N}}$ is a valid input for FiniteSubcover, M would be required to produce some $k \in \mathbb{N}$ with $k \geqslant K$ rather than N. The contradiction can only be resolved by the assumption that M never completes an output on input $(U)_{n\in\mathbb{N}}$ for $U \neq X$, which yields the final answer to be $\bot \in \mathbb{S}$.

(6) ⇔ (7) This follows via Proposition 4.2(1).

(1) ⇒ (8) By Proposition 3.3(2), we can compute $y \mapsto (x \mapsto A(x, y))$ from $A \in \mathcal{A}(\mathbf{X}\times\mathbf{Y})$. Now we find $(x \mapsto A(x, y)) \in \mathcal{A}(\mathbf{X})$, and if $\text{IsEmpty}_{\mathbf{X}}$ is computable, so is $y \mapsto \text{IsEmpty}_{\mathbf{X}}(x \mapsto A(x, y)) = \pi_2(A)$.

(8) ⇒ (9) Just instantiate with $\mathbf{Y} = \mathbb{N}$.

(9) ⇒ (1) Let $y \in \mathbf{Y}$ be a (computable) point. Then $A \mapsto (y \in \pi_2(A \times Y)^C)$ realizes $\text{IsEmpty}_{\mathbf{X}}$, and is continuous (computable) using the assumption, together with Proposition 4.2(1, 6, 7) together with Corollary 3.4. □

Proposition 5.3. *Let* $\mathbf{X}$ *be* (*computably*) *compact and* $f : \mathbf{X} \to \mathbf{Y}$ *a* (*computable*) *continuous surjection. Then* $\mathbf{Y}$ *is* (*computably*) *compact.*

Proof. We use the characterization of compactness provided by Proposition 5.2(2). By Proposition 4.2(6) we find $f^{-1} : \mathcal{O}(\mathbf{Y}) \to \mathcal{O}(\mathbf{X})$ to be continuous (computable). Now observe $\text{IsFull}_{\mathbf{Y}}(U) = \text{IsFull}_{\mathbf{X}}(f^{-1}(U))$ due to surjectivity of f. □

Proposition 5.4. *If* $\mathbf{X}$, $\mathbf{Y}$ *are* (*computably*) *compact, then so is* $\mathbf{X} \times \mathbf{Y}$.

Proof. Note $\text{IsEmpty}_{\mathbf{X}\times\mathbf{Y}} = \text{IsEmpty}_{\mathbf{Y}} \circ \pi_2$, and use Proposition 5.2(8). □

We can call a subset of a represented space compact, if it is compact when represented with the subspace representation. A set $K \subseteq \mathbf{X}$ is called *saturated*, iff $K = \bigcap_{\{U\in\mathcal{O}(\mathbf{X}) \mid K\subseteq U\}} U$. Noting that for compact K the set $\{U \in \mathcal{O}(\mathbf{X}) \mid K \subseteq U\}$ is open in $\mathcal{O}(\mathbf{X})$, we obtain a representation of the set $\mathcal{K}(\mathbf{X})$ of saturated compact sets by identifying it as a subspace of $\mathcal{O}(\mathcal{O}(\mathbf{X}))$. In particular, this makes $\text{IsContainedIn} : \mathcal{K}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ computable, and provides a uniform counterpart to the results in 5.2. For arbitrary sets $K \subseteq X$, we use $\uparrow K := \bigcap_{\{U\in\mathcal{O}(\mathbf{X}) \mid K\subseteq U\}} U$ to denote its *saturation*, and point out that any $\uparrow K$ is saturated. In a T_1 space, i.e. a represented space $\mathbf{X}$ where $\forall x \in \mathbf{X}$ $\{x\} \in \mathcal{A}(\mathbf{X})$, any set is already saturated.

We proceed to exhibit a number of computable operations on spaces of (saturated) compact sets, some of which can be seen as uniform counterparts of results about compact spaces above.

Proposition 5.5. *The following operations are well-defined and computable*:

(1) $\text{IsContainedIn} : \mathcal{K}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$.
(2) $x \mapsto \uparrow\{x\} : \mathbf{X} \to \mathcal{K}(\mathbf{X})$.
(3) $\cup : \mathcal{K}(\mathbf{X}) \times \mathcal{K}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$.
(4) $\uparrow\cap : \mathcal{K}(\mathbf{X}) \times \mathcal{A}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$.
(5) $f^{-1} \mapsto \uparrow f :\subseteq \mathcal{C}(\mathcal{O}(\mathbf{Y}), \mathcal{O}(\mathbf{X})) \to \mathcal{C}(\mathcal{K}(\mathbf{X}), \mathcal{K}(\mathbf{Y}))$; *here and below* $\uparrow f$ *is the composition of* f *and the saturation operator* $\uparrow$.
(6) $f \mapsto \uparrow f : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \to \mathcal{C}(\mathcal{K}(\mathbf{X}), \mathcal{K}(\mathbf{Y}))$.
(7) $(f, K) \mapsto \uparrow f[K] : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \times \mathcal{K}(\mathbf{X}) \to \mathcal{K}(\mathbf{Y})$.
(8) $\times : \mathcal{K}(\mathbf{X}) \times \mathcal{K}(\mathbf{Y}) \to \mathcal{K}(\mathbf{X} \times \mathbf{Y})$.
(9) $\pi_1 : \mathcal{K}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{K}(\mathbf{X})$, $\pi_2 : \mathcal{K}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{K}(\mathbf{Y})$.

Proof. (1) Taking into account the definition of $\mathcal{K}$, this is an instantiation of eval from Proposition 3.3(1).
(2) Note that $x \in U$ iff $\uparrow\{x\} \subseteq U$ for $U \in \mathcal{O}(\mathbf{X})$.
(3) $\cup : \mathcal{K}(\mathbf{X}) \times \mathcal{K}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$ is realized by $\cap : \mathcal{O}(\mathcal{O}(\mathbf{X})) \times \mathcal{O}(\mathcal{O}(\mathbf{X})) \to \mathcal{O}(\mathcal{O}(\mathbf{X}))$ from Proposition 4.2(2).
(4) The core observation is that $A \cap B \subseteq U$ iff $A \subseteq (U \cup B^C)$ together with Proposition 4.2(1, 2).
(5) It suffices to show that $f^{-1} \mapsto f$ is well-defined, its computability then follows from the computability of $f^{-1} \mapsto (f^{-1})^{-1} :\subseteq \mathcal{C}(\mathcal{O}(\mathbf{Y}), \mathcal{O}(\mathbf{X})) \to \mathcal{C}(\mathcal{O}(\mathcal{O}(\mathbf{X})), \mathcal{O}(\mathcal{O}(\mathbf{Y})))$ as it is a restriction of this map. The computability of the latter map in turn is a special case of Proposition 4.2(6). Well-definedness in turn follows directly from the characterization of $\mathcal{K}(\mathbf{X})$ as subspace of $\mathcal{O}(\mathcal{O}(\mathbf{X}))$.
(6) From (5) together with Proposition 4.2(6).
(7) From (6) via type conversion (Proposition 3.3).
(8) We show that given $A \in \mathcal{K}(\mathbf{X})$, $B \in \mathcal{K}(\mathbf{Y})$ and $U \in \mathcal{O}(\mathbf{X} \times \mathbf{Y})$, we can semidecide $A \times B \subseteq U$. Drawing from (1) and Proposition 4.2(9), we consider $y \mapsto \text{IsContainedIn}(A, \text{Cut}(y, U))$, which defines some element U_A of $\mathcal{O}(\mathbf{Y})$. Furthermore, we note that $A \times B \subseteq U$ iff $B \subseteq U_A$.
(9) By Proposition 3.1 the maps $\pi_1 : \mathbf{X} \times \mathbf{Y} \to \mathbf{X}$, $\pi_2 : \mathbf{X} \times \mathbf{Y} \to \mathbf{Y}$ are computable. Corollary 3.4 allows us to use (6) to lift this to compact sets. □

Corollary 5.6. $\mathbf{X}$ *is* (*computably*) *compact*, *iff* $\uparrow\text{id} : \mathcal{A}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$ *is well-defined and continuous* (*computable*).

Remarks. The represented space $\mathcal{K}(\mathbf{X})$ was introduced, and various parts of Proposition 5.5 were proven by Schröder in [57, Section 4.4.3] 2002 (for admissible $\mathbf{X}$). This definition of compactness also follows the approach of synthetic topology (Escardó 2004 [26]), and some of our results have also been obtained there (Proposition 5.2((2) $\Rightarrow$ (3)), Propositions 5.3, 5.4). The equivalence in Proposition 5.2((2) $\Leftrightarrow$ (8)) was shown by Escardó in [29].

That $\mathcal{K}(\mathbf{X})$ can be identified with a certain subspace of $\mathcal{O}(\mathcal{O}(\mathbf{X}))$ is the statement of the Hofmann–Mislove theorem (e.g. [32], see also the extension by Schröder in [62]).

The role of saturation as canonization operation for compact sets is already discussed by Collins [18]. Collins has stated the results of Proposition 5.5(2–4, 6, 7) as part of [18, Theorem 3.23 and Theorem 3.24]; Corollary 5.6 as [18, Theorem 3.31(4)]; and Proposition 5.2((1) $\Rightarrow$ (8)) as [18, Proposition 3.33]. His definition of compactness [18, Definition 3.28(4)] is equivalent to the one here. However, the claimed characterization of the computably compact spaces as the images of Cantor space computable functions in [18, Proposition 3.30] (stated without proof) is wrong. A counterexample (due to de Brecht, personal communication) is one-point compactification of $\mathbb{N}^{\mathbb{N}}$.[10] This also impacts [18, Theorem 3.32].

[10] In general, the one-point compactification of a represented space (X, δ) can be introduced as $(X \cup \{\bot\}, \delta_C)$ where $\delta_C(0^n 1 p) = \delta(p)$ and $\delta_C(0^{\mathbb{N}}) = \bot$. Any one-point compactification is computably compact, but can only have a total Cantor space representation if the original space was locally compact.

Some results in this section generalize known results in the far more restricted setting of computable metric spaces. A version of Proposition 5.2[(5) $\Leftrightarrow$ (6)] was proven by Brattka and Presser (2003 [15]). Weihrauch proved the restricted version of 5.5(7) in 2003 [70]. For computable metric spaces, compact compactness can also be characterized by the computability of the outer radius of closed sets (taking values in $\mathbb{R}_>$) as shown by Le Roux and the author as [39, Proposition 19].

The equivalences in Proposition 5.2((3) $\Leftrightarrow$ (6) $\Leftrightarrow$ (7) $\Leftrightarrow$ (8)) are uniform counterparts to well-known characterizations of compactness in topology.

The space $\mathcal{K}(\mathbf{X})$ corresponds to the upper Vietoris topology (which is often defined on the hyperspace of closed sets). This topology was introduced by Vietoris in 1922 [65]. Again, a general source for hyperspace topologies is [5].

6. T_2 separation

The T_2 separation axiom can, in the context of represented spaces, be understood equivalently as the property of a space making either inequality or the subspace of compact sets well-behaved. The strong connection between T_2 separation and compactness is somewhat reminiscent of the ultrafilter approach to topology, where compactness means each ultrafilter converges to at least one point, and T_2 that they converge to at most one point. Several of our equivalences will require the T_0 property to work, which we understand in a non-effective way to mean that $x \neq y$ implies $\uparrow\{x\} \neq \uparrow\{y\}$. We also need the (non-effective and non-uniform) T_1 property, which we understand to mean that if $x \neq y$, then $\uparrow\{x\} \cap \uparrow\{y\} = \emptyset$.

Definition 6.1. 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).

Proposition 6.2. *The following properties are equivalent for a represented space* $\mathbf{X}$:

(1) $\mathbf{X}$ *is* (*computably*) T_2.
(2) $\mathrm{id} : \mathcal{K}(\mathbf{X}) \to \mathcal{A}(\mathbf{X})$ *is well-defined and continuous* (*computable*), *and* $\mathbf{X}$ *is* T_0.
(3) $\cap : \mathcal{K}(\mathbf{X}) \times \mathcal{K}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$ *is well-defined and continuous* (*computable*), *and* $\mathbf{X}$ *is* T_1.
(4) $x \mapsto \uparrow\{x\} : \mathbf{X} \to \mathcal{A}(\mathbf{X})$ *is well-defined, injective and continuous* (*computable*).
(5) $\neq : \mathbf{X} \times \mathbf{X} \to \mathbb{S}$ *defined by* $\neq(x,x) = \bot$ *and* $\neq(x,y) = \top$ *otherwise is continuous* (*computable*).
(6) $\Delta_{\mathbf{X}} = \{(x,x) \mid x \in \mathbf{X}\} \in \mathcal{A}(\mathbf{X} \times \mathbf{X})$ (*is computable*[11]).
(7) $\mathrm{Graph} : \mathcal{C}(\mathbf{Y},\mathbf{X}) \to \mathcal{A}(\mathbf{Y} \times \mathbf{X})$ *is well-defined and continuous* (*computable*) *for any represented space* $\mathbf{Y}$.

Proof.

(1) $\Rightarrow$ (2) The map is identical to $K \mapsto (x \mapsto \mathrm{IsContainedIn}((K \cap \{x\}), \emptyset))$. To see that the map is continuous (computable) under the assumption that $x \mapsto \{x\} : \mathbf{X} \to \mathcal{A}(\mathbf{X})$ is continuous (computable), first use Proposition 5.5(4) to obtain $K \cap \{x\} \in \mathcal{K}(\mathbf{X})$, then Proposition 5.5(1). Being T_0 is an obvious consequence.

(1) $\wedge$ (2) $\Rightarrow$ (3) The continuity (computability) of the map in (3) is a direct consequence of the continuity (computability) of the map in (2) and Proposition 5.5(4). The T_1 property follows from the well-definedness of $x \mapsto \{x\} : \mathbf{X} \to \mathcal{A}(\mathbf{X})$.

(3) $\Rightarrow$ (2) Observe that $x \in K$ holds for a compact set K, if and only if $\uparrow\{x\} \subseteq K$ holds. The T_1-property allows us to strengthen this to $x \in K$ iff $\uparrow\{x\} \cap K \neq \emptyset$. Hence $\mathrm{id} : \mathcal{K}(\mathbf{X}) \to \mathcal{A}(\mathbf{X})$ is given by $K \mapsto (x \mapsto \mathrm{IsContainedIn}(K \cap \uparrow\{x\}, \emptyset))$, if intersection of compact sets is continuous (computable) (using Propositions 5.5(1)).

(2) $\Rightarrow$ (4) Continuity (computability) and well-definedness of the map in (4) follows immediately from continuity (computability) and well-definedness of the map in (2) and Proposition 5.5(2). Injectiveness of the map is equivalent to the T_0-property.

[11] Again, we remind the reader that we call $A \in \mathcal{A}(\mathbf{X})$ computable, if it is a computable name. This generalizes Weihrauch's notion of a co-c.e. closed set, not that of a computable closed set!

(4) $\Rightarrow$ (2) Similar to (3) $\Rightarrow$ (2): In $K \mapsto (x \mapsto \text{IsContainedIn}(K \cap \uparrow\{x\}, \emptyset))$ use $x \mapsto \uparrow\{x\} : \mathbf{X} \to \mathcal{A}(\mathbf{X})$, and the intersection from Proposition 5.5(4).

(2) $\Rightarrow$ (5) Given $x, y \in \mathbf{X}$, compute $\uparrow\{x\}, \uparrow\{y\} \in \mathcal{K}(\mathbf{X})$ by Proposition 5.5(2), and then $\uparrow\{x\}, \uparrow\{y\} \in \mathcal{A}(\mathbf{X})$ by the assumption. Now the claim follows from $x \neq y$ iff $x \notin \uparrow\{y\} \vee y \notin \uparrow\{x\}$. The equivalence holds due to the T_0 property, and the right hand side is computable due to Proposition 4.1.

(5) $\Rightarrow$ (1) By Proposition 3.3(2) we find $x \mapsto (y \mapsto \neq(x, y))$ to be continuous (computable), but this has the same realizers as $x \mapsto \{x\}$.

(5) $\Leftrightarrow$ (6) This is just a reformulation along the definition of $\mathcal{A}(\mathbf{X} \times \mathbf{X})$.

(5) $\Rightarrow$ (7) This follows from $\text{Graph}(f) = \{(y, x) \mid f(y) = x\} \in \mathcal{A}(\mathbf{Y} \times \mathbf{X})$; that $f \times \text{id} : \mathbf{Y} \times \mathbf{X} \to \mathbf{X} \times \mathbf{X}$ is available as a continuous function; and Proposition 4.2(1, 6).

(7) $\Rightarrow$ (6) Pick $\mathbf{Y} := \mathbf{X}$, and then consider $\text{Graph}(\text{id}_{\mathbf{X}})$. □

At the first glance, Definition 6.1 seems to be a prime candidate for T_1 separation rather than T_2 separation. In light of Proposition 6.2, it is clear that the denotation T_2 is justified, too. The reason to prefer the identification as T_2 rather than T_1 separation lies in the observation that a represented space is T_2, if and only if the induced topological space is sequentially T_2 (a topological space is sequentially T_2 iff its diagonal is sequentially closed (e.g. [62, Section 2.4]), so the claim is Proposition 6.2(6)). On the other hand, there are represented spaces not satisfying Definition 6.1 but inducing a T_1 topology.[12]

With both compactness and the computable T_2 property in place, we can proceed to discuss *computably proper maps*, i.e. those (continuous) maps where the preimages of compacts are compact. First, we present a counterpart to a classic result from topology:

Proposition 6.3. *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).*

Proof. By Proposition 6.2(2) the map $\text{id} : \mathcal{K}(\mathbf{Y}) \to \mathcal{A}(\mathbf{Y})$ is continuous (computable) due to the T_2-property for $\mathbf{Y}$. The map $(f, A) \mapsto f^{-1}(A) : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \times \mathcal{A}(\mathbf{Y}) \to \mathcal{A}(\mathbf{X})$ is computable by Proposition 4.2(1, 6), Proposition 3.3(1). By Corollary 5.6 the map $\text{id} : \mathcal{A}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$ is continuous (computable) due to the compactness of $\mathbf{X}$. Composition of these maps yields the claim. □

Proposition 6.4. *Let* $f : \mathbf{X} \to \mathbf{Y}$ *be surjective, continuous (computable) and (computably) proper, i.e. let* $f^{-1} : \mathcal{K}(\mathbf{Y}) \to \mathcal{K}(\mathbf{X})$ *be well-defined and continuous (computable). If* $\mathbf{X}$ *is (computably)* T_2 *and* $\mathbf{Y}$ *is* T_1, *then* $\mathbf{Y}$ *is (computably)* T_2.

Proof. We use the characterization of the T_2 property given in Proposition 6.2(3), i.e. the intersection of compact sets being computable as a compact set. For surjective f, we have $A \cap B = f[f^{-1}(A) \cap f^{-1}(B)]$. Using this equation for $A, B \in \mathcal{K}(\mathbf{Y})$, we can compute $f^{-1}(A), f^{-1}(B)$ by the assumption f were computably proper, then $f^{-1}(A) \cap f^{-1}(B)$ as a compact set by assumption $\mathbf{X}$ were T_2, and finally $\uparrow f[f^{-1}(A) \cap f^{-1}(B)] \in \mathcal{K}(\mathbf{Y})$ due to Proposition 5.5(6). Thus, we see that we can obtain $\uparrow(A \cap B) \in \mathcal{K}(\mathbf{Y})$ from $A, B \in \mathcal{K}(\mathbf{Y})$. By inspecting the proof of Proposition 6.2((3) $\Rightarrow$ (2)), we notice that we only care whether the intersection of two compact sets is empty or not. As saturation preserves the empty set, we conclude that computability of $\uparrow\cap : \mathcal{K}(\mathbf{Y}) \times \mathcal{K}(\mathbf{Y}) \to \mathcal{K}(\mathbf{Y})$ suffices for a T_1 space $\mathbf{Y}$ to be computably Hausdorff. □

Corollary 6.5. *If* $\mathbf{X}$ *is* T_1 *and has a computably proper representation* $\delta_{\mathbf{X}}$, *then* $\mathbf{X}$ *is computably* T_2.

[12] An example is the subspace $\{\{n\} \mid n \in \mathbb{N}\} \subseteq \mathcal{A}(\mathbb{N})$, as pointed out by Weihrauch in [72]. This space corresponds to the cofinite topology on $\mathbb{N}$.

Remarks. Escardó uses the condition Proposition 6.2(5) to introduce the Hausdorff property in [26]; and proceeds to prove Proposition 6.2((5) $\Rightarrow$ (2)). Weihrauch studied computable separation in countably based spaces in [72] (2010), including what corresponds to the equivalence Proposition 6.2((1) $\Rightarrow$ (5)).

Collins translated Escardó's definition of Hausdorff and the equivalence from Proposition 6.2((5) $\Rightarrow$ (2)) into computable analysis in [18, Definition 3.28(2) and Theorem 3.31(2)] (under the name *effectively distinguishable*).

Proper representations were studied by Schröder in [60] 2004; in particular, [60, Theorem 5.3] shows that having a proper admissible representation even implies metrizability (and is implied by metrizability, in turn); which is much stronger than Corollary 6.5. A minor further strengthening of the result can be found in [37], where it is shown that any such space embeds computably into a computable metric space.

7. Overtness

Like compactness, *overtness* both is a property of represented spaces and induces a space of certain subsets. Unlike compactness, overtness is classically vacuous and thus not that well-known. As discussed below, overtness has been present in the study of representations of spaces of subsets in computable analysis for a long time, although in a disguised rôle.

Definition 7.1. Let $\text{IsNonEmpty}_{\mathbf{X}} : \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ be defined by $\text{IsNonEmpty}_{\mathbf{X}}(\emptyset) = \bot$ and $\text{IsNonEmpty}_{\mathbf{X}}(U) = \top$ for $U \neq \emptyset$. Now call $\mathbf{X}$ (computably) *overt*, iff $\text{IsNonEmpty}_{\mathbf{X}}$ is continuous (computable).

This definition shows that overtness is in some sense the dual to compactness. In computable analysis, typically the criterion separable rather than overt has been used for spaces, based on the following observation:

Proposition 7.2. *Let $(a_n)_{n\in\mathbb{N}}$ be a dense sequence in $\mathbf{X}$. Then $\text{IsNonEmpty}_{\mathbf{X}}$ is computable relative to $(a_n)_{n\in\mathbb{N}}$.*

Proof. If $U \in \mathcal{O}(\mathbf{X})$ is non-empty, it contains some a_k. Thus simultaneously testing $a_i \in U$ for all $i \in \mathbb{N}$ will detect non-emptyness of U, if true. $\square$

As all represented spaces admit a dense sequence (by lifting a dense sequence in the domain of the representation), we see that all represented spaces are overt – merely computable overtness survives as a distinguishing criterion.

Proposition 7.3. *Let $\mathbf{X}$ be (computably) overt and $\mathbf{Y}$ be (computably) T_2. Then $\mathcal{C}(\mathbf{X}, \mathbf{Y})$ is (computably) T_2.*

Proof. For $f, g : \mathbf{X} \to \mathbf{Y}$ we find $f \neq g \Leftrightarrow \text{IsNonEmpty}_{\mathbf{X}}(\{x \in \mathbf{X} \mid f(x) \neq g(x)\})$. $\square$

Similar to our procedure for compact sets, we can introduce the space $\mathcal{V}(\mathbf{X})$ of overt sets. The crucial mapping will be $\text{Intersects} : \mathcal{V}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ defined by $\text{Intersects}(A, U) = \top$ iff $A \cap U \neq \emptyset$. This makes clear that $\mathcal{V}(\mathbf{X})$ can be understood as a subspace of $\mathcal{O}(\mathcal{O}(\mathbf{X}))$. A set $O \in \mathcal{O}(\mathcal{O}(\mathbf{X}))$ will correspond to an element of $\mathcal{V}(\mathbf{X})$, iff it is of the form $O_A = \{U \in \mathcal{O}(\mathbf{X}) \mid U \cap A \neq \emptyset\}$ for some subset $A \subseteq X$. Note that $O_A = O_B$ for two sets $A, B \subseteq X$, iff $\overline{A} = \overline{B}$ (here $\overline{}$ denotes the topological closure). Hence, we obtain $\mathcal{V}(\mathbf{X})$ by interpreting each set of the form $O_A \in \mathcal{O}(\mathcal{O}(\mathbf{X}))$ as $\overline{A} \subseteq X$.

In particular, we see that the overt sets and the closed sets coincide extensionally. In fact, the equivalence class of representations for $\mathcal{V}(\mathbf{X})$ has been studied for a long time, called the representation of closed sets by positive information. It is known that this representation is incomparable to the representation by negative information, i.e. that neither $\text{id} : \mathcal{V}(\mathbf{X}) \to \mathcal{A}(\mathbf{X})$ nor $\text{id} : \mathcal{A}(\mathbf{X}) \to \mathcal{V}(\mathbf{X})$ is continuous for non-empty $\mathbf{X}$. Our justification to avoid the word *closed* for the space $\mathcal{V}(\mathbf{X})$ lies in the computable structure available on it, which differs significantly from the usual closure properties expected from closed sets:

Proposition 7.4. *Let $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ be represented spaces, and let $\mathbf{Z}$ be computably overt. Then the following functions are computable:*

(1) $\overline{} : \mathcal{O}(\mathbf{Z}) \to \mathcal{V}(\mathbf{Z})$.
(2) $x \mapsto \overline{\{x\}} : \mathbf{X} \to \mathcal{V}(\mathbf{X})$.
(3) $\cup : \mathcal{V}(\mathbf{X}) \times \mathcal{V}(\mathbf{X}) \to \mathcal{V}(\mathbf{X})$.
(4) $\overline{\bigcup} : \mathcal{C}(\mathbb{N}, \mathcal{V}(\mathbf{X})) \to \mathcal{V}(\mathbf{X})$.
(5) $\overline{\cap} : \mathcal{V}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathcal{V}(\mathbf{X})$.
(6) $\overline{\pi_1} : \mathcal{V}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{V}(\mathbf{X})$.
(7) $(f, A) \mapsto \overline{f[A]} : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \times \mathcal{V}(\mathbf{X}) \to \mathcal{V}(\mathbf{Y})$.
(8) $(f^{-1}, A) \mapsto \overline{f[A]} :\subseteq \mathcal{C}(\mathcal{O}(\mathbf{Y}), \mathcal{O}(\mathbf{X})) \times \mathcal{V}(\mathbf{X}) \to \mathcal{V}(\mathbf{Y})$, *which takes a function between open sets that is the preimage map obtained from some (topologically continuous) function* $f : \mathbf{X} \to \mathbf{Y}$ *and an overt subset* A *of* $\mathbf{X}$, *and outputs the closure of the image of* A *under* f *as an overt subset of* $\mathbf{Y}$.

Proof. (1) $\text{Intersects}(\overline{V}, U) = \text{IsNonEmpty}_{\mathbf{Z}}(U \cap V)$, using Proposition 4.2(3) for $\cap : \mathcal{O}(\mathbf{Z}) \times \mathcal{O}(\mathbf{Z}) \to \mathcal{O}(\mathbf{Z})$.
(2) $\text{Intersects}(\overline{\{x\}}, U) = (x \in U)$, using Proposition 4.2(7) for $\in : \mathbf{X} \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$.
(3) $\text{Intersects}(\overline{A \cup B}, U) = \vee(\text{Intersects}(A, U), \text{Intersects}(B, U))$.
(4) $\text{Intersects}(\overline{\bigcup_{n\in\mathbb{N}} A_n}, U) = \bigvee_{n\in\mathbb{N}}(\text{Intersects}(A_n, U))$.
(5) $\text{Intersects}(\overline{V \cap A}, U) = \text{Intersects}(A, V \cap U)$, using Proposition 4.2(3) for $\cap : \mathcal{O}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathcal{O}(\mathbf{X})$.
(6) $\text{Intersects}(\overline{\pi_1(A)}, U) = \text{Intersects}(A, U \times Y)$, using Proposition 4.2(8) for $\times : \mathcal{O}(\mathbf{X}) \times \mathcal{O}(\mathbf{Y}) \to \mathcal{O}(\mathbf{X} \times \mathbf{Y})$.
(7) $\text{Intersects}(\overline{f[A]}, U) = \text{Intersects}(A, f^{-1}(U))$, using Proposition 4.2(6) to obtain $f^{-1} : \mathcal{O}(\mathbf{Y}) \to \mathcal{O}(\mathbf{X})$.
(8) $\text{Intersects}(\overline{f[A]}, U) = \text{Intersects}(A, f^{-1}(U))$. □

Remarks. The notion of overtness was named as such by Taylor, the definition here follows Escardó [26]. This reference also contains Proposition 7.3 and Proposition 7.4(7). The statements of Proposition 7.2 and 7.4(5) are very similar to [4, Proposition 2.2] by Bauer and Lesnik (2012). In restricted settings such as Euclidean spaces, Weihrauch has studied $\mathcal{V}(\mathbf{X})$ as the closed subsets of $\mathbf{X}$ represented by positive information, and e.g. proven Proposition 7.4(3) in [69]. The representation of $\mathcal{V}(\mathbf{X})$ via an identification with a subspace of $\mathcal{O}(\mathcal{O}(\mathbf{X}))$ was done by Schröder in [57, Section 4.4.2] (2002), together with several of its closure properties.

Both Escardó and Taylor have pointed out that classically, every space is overt.

Collins translated the definition of overtness into the language of computable analysis (under the name *effectively separable*), and introduced the notation $\mathcal{V}(\mathbf{X})$ in [18] (2010). He discusses the role of topological closure as the canonization operation for the overt sets; and provides the results of Proposition 7.4(1–5, 7). His claimed characterization of the computably overt spaces as those having a computable dense sequence in [18, Proposition 3.30] (stated without proof) is wrong, though. A counterexample is the space $\{p \in \{0, 1\}^{\mathbb{N}} \mid p \text{ is not computable}\}$ understood as a subspace of $\{0, 1\}^{\mathbb{N}}$. The failure of [18, Proposition 3.30] then impacts [18, Theorem 3.32].

The space $\mathcal{V}(\mathbf{X})$ corresponds to the lower Vietoris topology (or, equivalently, the lower Fell topology) (as shown by Schröder). This topology was introduced by Vietoris in 1922 [65]. Again, a general source for hyperspace topologies is [5]. Thus, we can obtain the Vietoris topology on a space of subsets as $\mathcal{K}(\mathbf{X}) \wedge \mathcal{V}(\mathbf{X})$ and the Fell topology as $\mathcal{A}(\mathbf{X}) \wedge \mathcal{V}(\mathbf{X})$. Both of these spaces appear in a variety of results.

8. Discreteness

Just as the T_2 separation property could be characterizes as making intersection of compact sets computable, there is a property of a space that makes intersection of overt sets computable. This property, *discreteness*, is in many ways the dual to T_2 separation, just as overt is the dual to compact.

Definition 8.1. A represented space $\mathbf{X}$ is called (computably) discrete, iff the map $x \mapsto \{x\} : \mathbf{X} \to \mathcal{O}(\mathbf{X})$ is well-defined and continuous (computable).

Theorem 8.2. *The following properties are equivalent for a represented space* $\mathbf{X}$:

(1) $\mathbf{X}$ *is (computably) discrete.*

(2) $\mathrm{id} : \mathcal{V}(\mathbf{X}) \to \mathcal{O}(\mathbf{X})$ *is well-defined and continuous* (*computable*), *and* $\mathbf{X}$ *is* T_1.
(3) $\cap : \mathcal{V}(\mathbf{X}) \times \mathcal{V}(\mathbf{X}) \to \mathcal{V}(\mathbf{X})$ *is well-defined and continuous* (*computable*), *and* $\mathbf{X}$ *is* T_1.
(4) $= : \mathbf{X} \times \mathbf{X} \to \mathbb{S}$ *defined by* $=(x, x) = \top$ *and* $=(x, y) = \bot$ *otherwise is continuous* (*computable*).
(5) $\Delta_{\mathbf{X}} = \{(x, x) \mid x \in \mathbf{X}\} \in \mathcal{O}(\mathbf{X} \times \mathbf{X})$ (*is computable*).

Proof. (1) $\Rightarrow$ (2) We find $x \in A$ iff $\mathrm{Intersects}(A, \{x\})$. The (topological) T_1 property is a straightforward consequence of singletons being open.
(2) $\Rightarrow$ (3) Use $\overline{\cap}$ from Proposition 7.4(5) together with the assumption.
(3) $\Rightarrow$ (4) Given $(x, y) \in \mathbf{X} \times \mathbf{X}$, we can use Proposition 7.4(2) to compute $(\overline{\{x\}}, \overline{\{y\}}) \in \mathcal{V}(\mathbf{X}) \times \mathcal{V}(\mathbf{X})$. The T_1 property means $\overline{\{x\}} = \{x\}$ and $\overline{\{y\}} = \{y\}$. Subsequently we use $\cap$ from the assumption to compute $\{x\} \cap \{y\} \in \mathcal{V}(\mathbf{X})$, and then $\mathrm{Intersects}(\{x\} \cap \{y\}, X)$, which is identical to $=(x, y)$.
(4) $\Rightarrow$ (1) This is a consequence of currying to $x \mapsto (y \mapsto =(x, y))$.
(4) $\Leftrightarrow$ (5) This is straightforward using $\mathcal{C}(\mathbf{X} \times \mathbf{X}, \mathbb{S}) \cong \mathcal{O}(\mathbf{X} \times \mathbf{X})$. □

As any represented space is separable (as pointed out in Section 7), the requirement that $\{x\} \in \mathcal{O}(\mathbf{X})$ for any $x \in \mathbf{X}$ already forces $\mathbf{X}$ to be countable, thus producing the following:

Proposition 8.3. *Any discrete represented space is countable.*

It is worth pointed out that – unlike in topology – being computably discrete does not imply being computably Hausdorff. A counterexample can be constructed by starting with a recursively enumerable but not recursive set $A \subseteq \mathbb{N}$, and then identifying the elements of A. This means we define a representation $\delta : \mathbb{N}^{\mathbb{N}} \to (\mathbb{N} \setminus A) \cup \{A\}$ by $\delta(p) = A$ if $p(0) \in A$ and $\delta(p) = p(0)$ if $p(0) \notin A$. This makes equality recognizable, but not refutable.

Proposition 8.4. *Let* $\mathbf{X}$ *be* (*computably*) *compact and* $\mathbf{Y}$ *be* (*computably*) *discrete. Then* $\mathcal{C}(\mathbf{X}, \mathbf{Y})$ *is* (*computably*) *discrete.*

Proof. We find $=(f, g) = \mathrm{IsFull}_{\mathbf{X}}(\{x \in \mathbf{X} \mid =(f(x), g(x))\})$, the claim now follows with Theorem 8.2(4) and Proposition 5.2(2). □

Unlike for its dual result in Proposition 7.3, it is easy to find a counter-example showing the necessity of a restriction on the domain in the preceding result. As such, consider $\mathbf{X} = \mathbf{Y} = \mathbb{N}$. The space $\mathbb{N}$ is computably discrete, but not compact, and $\mathcal{C}(\mathbb{N}, \mathbb{N}) \cong \mathbb{N}^{\mathbb{N}}$ is not discrete by Proposition 8.3.

Remarks. Escardó uses the characterization in Theorem 8.2(4) as definition of discreteness, and proceeds to prove Theorem 8.2((4) $\Rightarrow$ (2)) and Proposition 8.4 in [26]. That computable discreteness does not imply computably Hausdorff has been observed by both Escardó; and by Weihrauch in [72].

In the setting of computable analysis, Collins introduced discreteness and proved the equivalence from Theorem 8.2((1) $\Leftrightarrow$ (2)) as [18, Theorem 3.31(1)].

9. Admissibility as effective T_0 separation

Recall the computable maps $x \mapsto \uparrow\{x\} : \mathbf{X} \to \mathcal{K}(\mathbf{X})$ (Proposition 5.5(2)) and $x \mapsto \overline{\{x\}} : \mathbf{X} \to \mathcal{V}(\mathbf{X})$ (Proposition 7.4(2)), and note that both are realized in the same way: An element $x \in \mathbf{X}$ provides the abstract capacity to determine membership in an open set (since $x \in U \Leftrightarrow \uparrow\{x\} \subseteq U \Leftrightarrow \overline{\{x\}} \cap U \neq \emptyset$).

Definition 9.1. Consider the computable map $\kappa_{\mathbf{X}} : \mathbf{X} \to \mathcal{O}(\mathcal{O}(\mathbf{X}))$ defined by $\kappa_{\mathbf{X}}(x) = \{U \in \mathcal{O}(\mathbf{X}) \mid x \in U\}$. Let $\mathbf{X}_\kappa$ denote the image of $\mathbf{X}$ in $\mathcal{O}(\mathcal{O}(\mathbf{X}))$ under $\kappa_{\mathbf{X}}$ (equipped with the subspace representation).

With the considerations above, we see that we may consider $\mathbf{X}_\kappa$ simultaneously as a subspace of $\mathcal{K}(\mathbf{X})$ and $\mathcal{V}(\mathbf{X})$. If we understand $\mathbf{X}_\kappa$ as a subspace of $\mathcal{K}(\mathbf{X})$, then $\kappa_\mathbf{X}(x) = \uparrow\{x\}$, if as a subspace of $\mathcal{V}(\mathbf{X})$, then $\kappa_\mathbf{X}(x) = \overline{\{x\}}$. This identification will be helpful to see that κ is an endofunctor[13] on the category-extension of represented spaces. That κ commutes with composition is rather obvious, hence we only need the following:

Proposition 9.2. *There is an induced computable map* $\kappa : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \to \mathcal{C}(\mathbf{X}_\kappa, \mathbf{Y}_\kappa)$ *for all represented spaces* $\mathbf{X}, \mathbf{Y}$ *such that the following diagram commutes*:

$$\begin{tikzcd}
\mathbf{X} \arrow[r, "f"] \arrow[d, "\kappa_\mathbf{X}"] & \mathbf{Y} \arrow[d, "\kappa_\mathbf{Y}"] \\
\mathbf{X}_\kappa \arrow[r, "\kappa(f)"] & \mathbf{Y}_\kappa
\end{tikzcd}$$

Proof. This follows e.g. from Proposition 5.5(6) and $\mathbf{X}_\kappa \subseteq \mathcal{K}(\mathbf{X})$, together with $\uparrow f[\uparrow\{x\}] = \uparrow\{f(x)\}$. □

In general, $\kappa_\mathbf{X} : \mathbf{X} \to \mathbf{X}_\kappa$ will fail to be (computably) continuously invertible, however, those spaces admitting a continuous (computable) left-inverse for $\kappa_\mathbf{X}$ can be characterized as exactly those fully understandable in terms of their topology.

Definition 9.3. Call $\mathbf{X}$ (computably) admissible, iff $\kappa_\mathbf{X} : \mathbf{X} \to \mathbf{X}_\kappa$ admits a continuous (computable) left-inverse.

If $\mathbf{X}$ is computably admissible, then $\kappa_\mathbf{X} : \mathbf{X} \to \mathbf{X}_\kappa$ is even a computable isomorphism: By definition of $\mathbf{X}_\kappa$, we find that $\kappa_\mathbf{X}$ is surjective; and in the category of represented spaces we have that any surjective section is an isomorphism.

We will proceed to demonstrate that κ is the reflector of the (Cartesian-closed) subcategory of the (computably) admissible represented spaces inside the category of represented spaces. In a way, this reflector is analogous to the Kolmogorov-quotient (or T_0-reflector) in topology.

Corollary 9.4. *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* $f = \mathfrak{R}(f) \circ \kappa_\mathbf{X}$ *for all* $f \in \mathcal{C}(\mathbf{X}, \mathbf{Y})$.

Proof. Take the map $\kappa : \mathcal{C}(\mathbf{X}, \mathbf{Y}) \to \mathcal{C}(\mathbf{X}_\kappa, \mathbf{Y}_\kappa)$ from Proposition 9.2. Then we use Proposition 3.3(4, 6) to compose $\kappa(f)$ with $\kappa_\mathbf{Y}^{-1}$ obtained from Definition 9.3. □

Note that in Corollary 9.4 each function $\mathfrak{R}(f) : \mathbf{X}_\kappa \to \mathbf{Y}$ is uniquely determined by $f = \mathfrak{R}(f) \circ \kappa_\mathbf{X}$, as $\kappa_\mathbf{X}$ is surjective.

Proposition 9.5. $\mathbb{S}$ *is computably admissible.*

Proof. We can explicitly define $\kappa_\mathbb{S}^{-1} :\subseteq \mathcal{O}(\mathcal{O}(\mathbb{S})) \to \mathbb{S}$ by $\kappa_\mathbb{S}^{-1}(U) = (\{\top\} \in U)$. □

Theorem 9.6. *Let* $\mathbf{Y}$ *be (computably) admissible. Then* $\mathcal{C}(\mathbf{X}, \mathbf{Y})$ *is (computably) admissible.*

Proof. Given some $U \in \mathcal{O}(\mathbf{Y})$ and $x \in \mathbf{X}$, we can compute $\{f \in \mathcal{C}(\mathbf{X}, \mathbf{Y}) \mid f(x) \in U\} \in \mathcal{O}(\mathcal{C}(\mathbf{X}, \mathbf{Y}))$. Hence, given $x \in \mathbf{X}$ and $\{U \in \mathcal{O}(\mathcal{C}(\mathbf{X}, \mathbf{Y})) \mid f \in U\} \in \mathcal{O}(\mathcal{O}(\mathcal{C}(\mathbf{X}, \mathbf{Y})))$ we can compute $\{U \in \mathcal{O}(\mathbf{Y}) \mid f(x) \in U\} \in \mathcal{O}(\mathcal{O}(\mathbf{Y}))$ by Proposition 4.2(6). By assumption, the latter suffices to compute $f(x) \in \mathbf{Y}$. Currying yields the claim. □

Corollary 9.7. $\mathbf{X}_\kappa$, $\mathcal{O}(\mathbf{X})$, $\mathcal{A}(\mathbf{X})$, $\mathcal{K}(\mathbf{X})$ *and* $\mathcal{V}(\mathbf{X})$ *are all computably admissible.*

[13]In fact, we could even call κ a *computable* endofunctor, as its restriction to any homset is computable.

Observation 9.1. Any subspace of a (computably) admissible space is (computably) admissible.

Corollary 9.8. *$\mathcal{O}(\mathbf{X}) \cong \mathcal{O}(\mathbf{X}_\kappa)$, $\mathcal{A}(\mathbf{X}) \cong \mathcal{A}(\mathbf{X}_\kappa)$, $\mathcal{K}(\mathbf{X}) \cong \mathcal{K}(\mathbf{X}_\kappa)$ and $\mathcal{V}(\mathbf{X}) \cong \mathcal{V}(\mathbf{X}_\kappa)$.*

Corollary 9.9. *$\kappa_\mathbf{X}$ is effectively open, i.e. there is a well-defined and computable map $K : \mathcal{O}(\mathbf{X}) \to \mathcal{O}(\mathbf{X}_\kappa)$ such that $K(U) = \kappa_\mathbf{X}[U]$.*

Corollary 9.10. *$\mathbf{X}$ is (computably) compact, (computably) T_2, (computably) overt or (computably) discrete if and only if $\mathbf{X}_\kappa$ has that property.*

Theorem 9.11. *The following properties are equivalent for a represented space $\mathbf{X}$:*

(1) *$\mathbf{X}$ is (computably) admissible.*
(2) *$f \mapsto f_K : \mathcal{C}(\mathbf{Y}, \mathbf{X}) \to \mathcal{C}(\mathcal{K}(\mathbf{Y}), \mathcal{K}(\mathbf{X}))$ has a well-defined and continuous (computable) partial inverse for any represented space $\mathbf{Y}$; where $f_K(A) = \uparrow f[A]$.*
(3) *$f \mapsto f^{-1} : \mathcal{C}(\mathbf{Y}, \mathbf{X}) \to \mathcal{C}(\mathcal{O}(\mathbf{X}), \mathcal{O}(\mathbf{Y}))$ has a well-defined and continuous (computable) partial inverse for any represented space $\mathbf{Y}$.*
(4) *Any topologically continuous function $f : \mathbf{Y} \to \mathbf{X}$ (i.e. $f^{-1} : \mathcal{O}(\mathbf{X}) \to \mathcal{O}(\mathbf{Y})$ is well-defined) is continuous as a function between represented spaces (i.e. $f \in \mathcal{C}(\mathbf{Y}, \mathbf{X})$) (no computable counterpart).*

Proof. (1) ⇒ (2) For precision, let f_K be the input, i.e. the map from compact sets to compact sets derived from the desired output f. Now note $f = \kappa_\mathbf{X}^{-1} \circ f_K \circ \kappa_\mathbf{Y}$, and consider Proposition 9.2, Definition 9.3 and Proposition 3.3(4).

(2) ⇒ (3) Given a continuous function of the form $f^{-1} : \mathcal{O}(\mathbf{X}) \to \mathcal{O}(\mathbf{Y})$ induced by some continuous $f : \mathbf{Y} \to \mathbf{X}$, we may use Proposition 5.5(4) to obtain the induced function $f_K : \mathcal{K}(\mathbf{Y}) \to \mathcal{K}(\mathbf{X})$. By assumption, we can recover $f \in \mathcal{C}(\mathbf{X}, \mathbf{Y})$ from the latter.

(3) ⇒ (1) First we shall show that $\kappa_\mathbf{X}^{-1} : \mathbf{X}_\kappa \to \mathbf{X}$ is well-defined, using proof-by-contradiction. So let us assume $\kappa_\mathbf{X}$ were not injective, i.e. there were $x \neq y \in \mathbf{X}$ with $\kappa_\mathbf{X}(x) = \kappa_\mathbf{X}(y)$, hence $x \in U \Leftrightarrow y \in U$ for any $U \in \mathcal{O}(\mathbf{X})$. Then the constant function $\overline{x}, \overline{y} : \{0\} \to \mathbf{X}$ with $\overline{x}(0) = x$ and $\overline{y}(0) = y$ are continuous and distinct, but still $\overline{x}^{-1} = \overline{y}^{-1}$ as maps of the type $\mathcal{O}(\mathbf{X}) \to \mathcal{O}(\{0\})$. This contradicts the well-definedness of $f^{-1} \mapsto f :\subseteq \mathcal{C}(\mathcal{O}(\mathbf{X}), \mathcal{O}(\{0\})) \to \mathcal{C}(\{0\}, \mathbf{X})$.
Consider $\mathbf{Y} := \mathbf{X}_\kappa$. By Corollary 9.9, $\kappa_\mathbf{X}$ is effectively open, hence $\kappa_\mathbf{X} \in \mathcal{C}(\mathcal{O}(\mathbf{X}), \mathcal{O}(\mathbf{X}_\kappa))$ is a computable element. As $\kappa_\mathbf{X}^{-1}$ is well-defined, $\kappa_\mathbf{X} = (\kappa_\mathbf{X}^{-1})^{-1}$, and we can use the assumption to obtain $\kappa_\mathbf{X}^{-1} : \mathbf{X}_\kappa \to \mathbf{X}$ from $\kappa_\mathbf{X}$ as open map.

(1) ⇒ (4) By Corollary 9.8, a function $f : \mathbf{Y} \to \mathbf{X}$ is topologically continuous, iff it is so as a function $f : \mathbf{Y} \to \mathbf{X}_\kappa$. We may curry topologically continuous functions, too, and obtain $\chi_f : \mathbf{Y} \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ as a topologically continuous function defined by $\chi_f(y, U) = \top$ iff $f(y) \in U$. By composition, so is $\chi_f \circ (\delta_\mathbf{Y} \times \delta_{\mathcal{O}(\mathbf{X})}) :\subseteq \{0,1\}^\mathbb{N} \to \mathbb{S}$. But this just means $\chi_f \circ (\delta_\mathbf{Y} \times \delta_{\mathcal{O}(\mathbf{X})}) \in \mathcal{O}(\operatorname{dom}(\delta_\mathbf{Y} \times \delta_{\mathcal{O}(\mathbf{X})}))$. Now the process can be reversed, using represented space continuity in place of topological continuity to find that $f \in \mathcal{C}(\mathbf{Y}, \mathbf{X}_\kappa)$. Admissibility of $\mathbf{X}$, i.e. $\mathbf{X} \cong \mathbf{X}_\kappa$, then allows us the make the final step and reach $f \in \mathcal{C}(\mathbf{Y}, \mathbf{X})$.

(4) ⇒ (1) Again, we first show that $\kappa_\mathbf{X}^{-1}$ is well-defined using proof-by-contradiction. Assume there were $x \neq y \in \mathbf{X}$ with $\kappa_\mathbf{X}(x) = \kappa_\mathbf{X}(y)$, hence $x \in U \Leftrightarrow y \in U$ for any $U \in \mathcal{O}(\mathbf{X})$. Then any function $f : \{0,1\}^\mathbb{N} \to \{x, y\} \subseteq \mathbf{X}$ is topologically continuous. However, there are $2^{2^{\aleph_0}}$ such functions, whereas $|\mathcal{C}(\{0,1\}^\mathbb{N}, \mathbf{X})| \leqslant 2^{\aleph_0}$.
Using again $\kappa_\mathbf{X} = (\kappa_\mathbf{X}^{-1})^{-1}$, as well as Corollary 9.9, we see that $\kappa_\mathbf{X}^{-1} : \mathbf{X}_\kappa \to \mathbf{X}$ is topologically continuous, hence we obtain it as a continuous function between represented spaces. □

Note that only the equivalence of (1), (2), (3) in the preceding theorem has a proper place in a synthetic treatment – the reference in 4. to the mere (external) well-definedness of f^{-1} is meaningless from a strictly internal view on a given category. Its provability here is due to the definition of continuity for represented spaces via (topological) continuity on Cantor space. What we obtain from it is the observation that the admissibly represented spaces (with continuous maps) simultaneously form a subcategory of the represented spaces and of the topological spaces, moreover, that this is in some sense the largest such joint subcategory.

As an example for the interplay of admissibility and some of the other properties of represented spaces, we shall briefly revisit the connection between a function and its graph (cf. Proposition 6.2(7)):

Proposition 9.12. *Let* $\mathbf{Y}$ *be (computably) admissible and (computably) compact. Then* $\mathrm{Graph}^{-1} :\subseteq \mathcal{A}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{C}(\mathbf{X}, \mathbf{Y})$ *is continuous (computable), where* $\mathrm{dom}(\mathrm{Graph}^{-1}) = \{A \in \mathcal{A}(\mathbf{X} \times \mathbf{Y}) \mid \exists f : \mathbf{X} \to \mathbf{Y}\ \mathrm{Graph}(f) = A\}$.

Proof. We may assume that $\mathrm{Graph}(f) \in \mathcal{A}(\mathbf{X} \times \mathbf{Y})$ and $x \in \mathbf{X}$ are given and show how to obtain $f(x) \in \mathbf{Y}$. Using Proposition 4.2(1, 9), we obtain $\{y \mid (x, y) \in \mathrm{Graph}(f)\} = \{f(x)\} \in \mathcal{A}(\mathbf{Y})$. Corollary 5.6 gets us $\uparrow\{f(x)\} \in \mathcal{K}(\mathbf{Y})$, by definition of $\mathbf{Y}_\kappa$ thus $\kappa_\mathbf{Y}(f(x)) \in \mathbf{Y}_\kappa$. By definition of admissibility, this in turn suffices to obtain $f(x) \in \mathbf{Y}$. □

Similar results can be obtained for compact and overt graphs:

Proposition 9.13. *Let* $\mathbf{Y}$ *be (computably) admissible and* $\mathbf{X}$ *(computably)* T_2*. Then* $\mathrm{Graph}^{-1} :\subseteq \mathcal{K}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{C}(\mathbf{X}, \mathbf{Y})$ *is continuous (computable), where* $\mathrm{dom}(\mathrm{Graph}^{-1}) = \{A \in \mathcal{K}(\mathbf{X} \times \mathbf{Y}) \mid \exists f : \mathbf{X} \to \mathbf{Y}\ \mathrm{Graph}(f) = A\}$.

Proof. We may assume that $\mathrm{Graph}(f) \in \mathcal{K}(\mathbf{X} \times \mathbf{Y})$ and $x \in \mathbf{X}$ are given and show how to obtain $f(x) \in \mathbf{Y}$. By Definition 6.1 we can obtain $\{x\} \in \mathcal{A}(\mathbf{X})$, and then $\{x\} \times Y \in \mathcal{A}(\mathbf{X} \times \mathbf{Y})$ by Proposition 4.2(8). Then Proposition 5.5(4) provides us with $\uparrow((\{x\} \times Y) \cap \mathrm{Graph}(f)) = \uparrow(\{x\} \times \{f(x)\}) \in \mathcal{K}(\mathbf{X} \times \mathbf{Y})$. Then we use projection (Proposition 5.5(9)) to get $\uparrow\{f(x)\} \in \mathcal{K}(\mathbf{Y})$, admissibility of $\mathbf{Y}$ enables us to extract $f(x)$. □

Proposition 9.14. *Let* $\mathbf{Y}$ *be (computably) admissible and* $\mathbf{X}$ *(computably) discrete. Then* $\mathrm{Graph}^{-1} :\subseteq \mathcal{V}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{C}(\mathbf{X}, \mathbf{Y})$ *is continuous (computable), where* $\mathrm{dom}(\mathrm{Graph}^{-1}) = \{A \in \mathcal{V}(\mathbf{X} \times \mathbf{Y}) \mid \exists f : \mathbf{X} \to \mathbf{Y}\ \mathrm{Graph}(f) = A\}$.

Proof. We may assume that $\mathrm{Graph}(f) \in \mathcal{V}(\mathbf{X} \times \mathbf{Y})$ and $x \in \mathbf{X}$ are given and show how to obtain $f(x) \in \mathbf{Y}$. By Definition 8.1 we can obtain $\{x\} \in \mathcal{O}(\mathbf{X})$, and then $\{x\} \times Y \in \mathcal{O}(\mathbf{X} \times \mathbf{Y})$ by Proposition 4.2(8). Then Proposition 7.4(5) provides us with $\overline{(\{x\} \times Y) \cap \mathrm{Graph}(f)} = \overline{\{x\} \times \{f(x)\}} \in \mathcal{V}(\mathbf{X} \times \mathbf{Y})$. Then we use projection (Proposition 7.4(6)) to get $\overline{\{f(x)\}} \in \mathcal{V}(\mathbf{Y})$, admissibility of $\mathbf{Y}$ enables us to extract $f(x)$. □

How to use admissibility. Admissibility is at the core of a very common scheme to prove computability of some specific mapping:

(1) Conclude that the solution set S is closed based on its definition (this often involves the Hausdorff condition).
(2) Obtain some compact candidate set K (either from the situation, or by assumption – this often involves some bounds).
(3) Compute $\uparrow(S \cap K)$ as a compact set via Proposition 5.5(4).
(4) Use domain-specific reasoning or assumption to conclude that the solution s is unique.
(5) As we have $\uparrow\{s\}$ available as a compact set, if our target space is admissible, we can compute s.

While this scheme is usually well-hidden and obscured, it is present e.g. in [31] by Galatolo, Hoyrup and Rojas, [55] by Rettinger, [20,21] by Collins and Graça. A very similar algorithm (albeit in a slightly different formal setting) is described by Escardó in [28]. This use of admissibility also underlies the model of *non-deterministic type-2 machines* suggest by Ziegler [77] and studied further by Brattka, de Brecht and the author in [11,14].

Remarks. The notion of admissibility was introduced by Kreitz and Weihrauch in 1985 [38]. They essentially define a represented space to be admissible if it is isomorphic to a subspace of $\mathcal{O}(\mathbb{N})$, which captures the countably based admissible spaces. Then they proceed to prove that this implies the condition in Theorem 9.11(4).

The understanding of admissibility as presented here is essentially due to Schröder. In [59] (2002) Schröder uses the condition in Theorem 9.11(4) as definition of admissibility, characterizes the topologies arising as $\mathcal{O}(\mathbf{X})$ and in particular proves a slightly weaker version of Theorem 9.6 (as it requires admissibility of $\mathbf{X}$, too). In his thesis [57] (2002), Schröder also defines computable admissibility (i.e. Definition 9.3) and provides the statements of Propositions 9.2, 9.5, Corollary 9.4 and Theorems 9.6, 9.11. Lietz also added to the development, e.g. [42, Theorem 3.2.7], which shows that Definition 9.3 is indeed well-suited for its purpose.

The identification of a point $x \in \mathbf{X}$ with its neighbourhood filter $\{U \in \mathcal{O}(\mathbf{X}) \mid x \in U\} \in \mathcal{O}(\mathcal{O}(\mathbf{X}))$ is reminiscent of the ultrafilter approach to topology.

The study of admissibility can be seen as an attempt to generalize the coincidence of computability relative to an oracle and (topological) continuity beyond the setting of Baire space. While dealing only with a restricted class of spaces, [13] by Brattka and Hertling (1994) and [54] by the author and Ziegler (2013) consider the questions for multivalued functions rather than just functions.

The connections between a continuous function and its graph as exemplified in Propositions 9.12, 9.13, 9.14 has been advanced by Brattka in [10] (2008).

10. Open relations and compact or overt sets

The dual nature of the spaces of overt and of compact sets becomes clearer when the interaction with open relations is studied. This in particular generalizes the well-known results about upper and lower computability of the maximum of compact subsets of the real numbers presented either as compact sets, or as overt sets (i.e. by positive information).

Proposition 10.1. *The map $\exists : \mathcal{O}(\mathbf{X} \times \mathbf{Y}) \times \mathcal{V}(\mathbf{X}) \to \mathcal{O}(\mathbf{Y})$ defined by $\exists(R, A) = \{y \in Y \mid \exists x \in A\ (x, y) \in R\}$ is computable. Moreover, whenever $\exists : \mathcal{O}(\mathbf{X} \times \mathbf{Y}) \times \mathcal{S}(\mathbf{X}) \to \mathcal{O}(\mathbf{Y})$ is computable for some hyperspace $\mathcal{S}(\mathbf{X})$ and some space $\mathbf{Y}$ containing a computable element y_0, then $\overline{} : \mathcal{S}(\mathbf{X}) \to \mathcal{V}(\mathbf{X})$ is computable.*

Proof. To see that $\exists$ is computable, note that $y \in \exists(R, A)$ iff $\text{Intersects}(A, \text{Cut}(y, R))$ and use Proposition 4.2(9) and the definition of $\mathcal{V}(\mathbf{X})$. Now if $\exists$ is computable for some other hyperspace $\mathcal{S}(\mathbf{X})$ in place of $\mathcal{V}(\mathbf{X})$, we can compute $\overline{} : \mathcal{S}(\mathbf{X}) \to \mathcal{V}(\mathbf{X})$ via $\text{Intersects}(\overline{A}, U) = \in(y_0, \exists(U \times Y, A))$. □

Corollary 10.2. $\bigcup : \mathcal{V}(\mathcal{O}(\mathbf{Y})) \to \mathcal{O}(\mathbf{Y})$ *is computable. (This functions maps $A \in \mathcal{V}(\mathcal{O}(\mathbf{Y}))$ to $(\bigcup_{U \in A} U) \in \mathcal{O}(\mathbf{Y})$.)*

Proof. Pick $\mathbf{X} = \mathcal{O}(\mathbf{Y})$, and instantiate R with $\{(U, x) \in \mathcal{O}(\mathbf{Y}) \times \mathbf{Y} \mid x \in U\}$. □

Proposition 10.3. *The map $\forall : \mathcal{O}(\mathbf{X} \times \mathbf{Y}) \times \mathcal{K}(\mathbf{X}) \to \mathcal{O}(\mathbf{Y})$ defined by $\forall(R, A) = \{y \in Y \mid \forall x \in A\ (x, y) \in R\}$ is computable. Moreover, whenever $\forall : \mathcal{O}(\mathbf{X} \times \mathbf{Y}) \times \mathcal{S}(\mathbf{X}) \to \mathcal{O}(\mathbf{Y})$ is computable for some hyperspace $\mathcal{S}(\mathbf{X})$ and some space $\mathbf{Y}$ containing a computable element y_0, then $\uparrow \text{id} : \mathcal{S}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$ is computable.*

Proof. To see that $\forall$ is computable, note that $y \in \forall(R, A)$ iff $\text{IsContainedIn}(A, \text{Cut}(y, R))$ and use Proposition 4.2(9) and Proposition 5.5(1). Now if $\forall$ is computable for some other hyperspace $\mathcal{S}(\mathbf{X})$ in place of $\mathcal{K}(\mathbf{X})$, we can compute $\uparrow \text{id} : \mathcal{S}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$ via $\text{IsContainedIn}(A, U) = \in(y_0, \forall(U \times Y, A))$. □

Corollary 10.4. $\bigcap : \mathcal{K}(\mathcal{O}(\mathbf{Y})) \to \mathcal{O}(\mathbf{Y})$ *is computable. (This function maps $A \in \mathcal{K}(\mathcal{O}(\mathbf{Y}))$ to $(\bigcap_{U \in A} U) \in \mathcal{O}(\mathbf{Y})$.)*

Proof. Pick $\mathbf{X} = \mathcal{O}(\mathbf{Y})$, and instantiate R with $\{(U, x) \in \mathcal{O}(\mathbf{Y}) \times \mathbf{Y} \mid x \in U\}$. □

To make the connection to the computability of maxima on the reals, we first generalize the Dedekind-cut construction of the reals. Let $\prec \in \mathcal{O}(\mathbf{X} \times \mathbf{X})$ be some transitive and open relation. Now we define $\overline{\mathbf{X}_\prec} := \{U \in \mathcal{O}(\mathbf{X}) \mid \forall x \in \mathbf{X}\ (\exists y \in U\ x \prec y \Rightarrow x \in U)\}$ as the subspace of $\mathcal{O}(\mathbf{X})$ containing the open initial segments of $\prec$. Next, consider the computable map $x \mapsto \text{Cut}(x, \prec) : \mathbf{X} \to \overline{\mathbf{X}_\prec}$ mapping x to $\{y \in \mathbf{X} \mid y \prec x\}$. We shall denote its image by $\mathbf{X}_\prec$, and identify an element x with its $\prec$-lower ideal. Depending on the properties of $\prec$, the map $\mathrm{id} : \mathbf{X} \to \mathbf{X}_\prec$ induced by such an identification may fail to be injective. Spaces of this form that have been studied so far are $\mathbb{R}_<$ and $\mathbb{R}_>$, the former in particular playing a central rôle in computable measure theory (e.g. Schröder [61], Collins [19]). By construction, the spaces $\mathbf{X}_\prec$ and $\overline{\mathbf{X}_\prec}$ are computably admissible.

Proposition 10.5.

(1) $\prec\, \in \mathcal{O}(\mathbf{X} \times \mathbf{X}_\prec)$, $\prec\, \in \mathcal{O}(\mathbf{X}_\succ \times \mathbf{X})$ *are computable.*
(2) $(x, y) \mapsto \{z \in \mathbf{X} \mid x \prec z \prec y\} : \mathbf{X}_\succ \times \mathbf{X}_\prec \to \mathcal{O}(\mathbf{X})$ *is computable.*
(3) *If* $\prec$ *is dense*[14] *and* $\mathbf{X}$ *is (computably) overt, then* $\prec\, \in \mathcal{O}(\mathbf{X}_\succ \times \mathbf{X}_\prec)$ *(is computable).*
(4) $\sup_\prec : \mathcal{V}(\mathbf{X}) \to \overline{\mathbf{X}_\prec}$ *and* $\sup_\prec : \mathcal{K}(\mathbf{X}) \to \overline{\mathbf{X}_\succ}$ *are computable.*
(5) *Let* $\mathbf{Y} \subseteq \mathbf{X}$ *be a dense*[15] *and computably overt subspace, and* $\prec$ *be dense. Then we can identify* $\overline{\mathbf{Y}_\prec}$ *and* $\overline{\mathbf{X}_\prec}$.

Proof. (1) $\prec : \mathbf{X} \times \mathbf{X}_\prec \to \mathbb{S}$ is a restriction of the computable map $\in : \mathbf{X} \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ from Proposition 4.2(7). The second claim follows by symmetry.
(2) This map is a restriction of the computable map $\cap : \mathcal{O}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathcal{O}(\mathbf{X})$ from Proposition 4.2(3).
(3) If $\prec$ is dense, then $x \prec y$ iff $\mathrm{IsNonEmpty}_\mathbf{X}(\{z \in \mathbf{X} \mid x \prec z \prec y\})$ which is computable by (2) and Definition 7.1.
(4) These are corollaries of Propositions 10.1, 10.3.
(5) The restriction of any element of $\overline{\mathbf{X}_\prec}$ to $\mathbf{Y}$ will yield an element of $\overline{\mathbf{Y}_\prec}$. Under the conditions given, we can recover a lower ideal $U \in \overline{\mathbf{X}_\prec}$ from its restriction to $\mathbf{Y}$ by noting $x \in U \Leftrightarrow \exists y \in \mathbf{Y}\ x \prec y \wedge y \in U$. □

If $\prec$ is e.g. a linear order, then the identification $\mathrm{id} : \mathbf{X} \to \mathbf{X}_\prec$ is injective, however, will usually not be computably invertible. The best we can obtain here is the following:

Proposition 10.6. *Let the intervals of* $\prec$ *be an effective base for* $\mathcal{O}(\mathbf{X})$*, i.e. let the computable map* $(a_i, b_i)_{i\in\mathbb{N}} \mapsto \bigcup_{i\in\mathbb{N}}\{x \mid a_i \prec x \prec b_i\} : \mathcal{C}(\mathbb{N}, \mathbf{X}\times\mathbf{X}) \to \mathcal{O}(\mathbf{X})$ *be surjective and admit a computable right-inverse (as a multivalued function). Then* $\mathbf{X}_\kappa \cong (\mathbf{X}_\prec \wedge \mathbf{X}_\succ)$.

Proof. As $\mathbf{X}_\prec$ and $\mathbf{X}_\succ$ are admissible, the identification maps $\mathrm{id} : \mathbf{X}_\kappa \to \mathbf{X}_\prec$ and $\mathrm{id} : \mathbf{X}_\kappa \to \mathbf{X}_\succ$ are well-defined and computable. It only remains to be shown that they admit a joint computable right-inverse. For this, we need to demonstrate how given $x \in \mathbf{X}_\prec \wedge \mathbf{X}_\succ$ and $U \in \mathcal{O}(\mathbf{X})$ we can recognize $x \in U$? By assumption, U can be effectively expressed as $\bigcup_{i\in\mathbb{N}}\{y \mid a_i \prec y \prec b_i\}$, and by Proposition 10.5(1) we can simultaneously semidecide $a_i \prec x$ and $x \prec b_i$, until for some $i \in \mathbb{N}$ both statements are true, in which case $x \in U$ is accepted. □

We just briefly introduce the represented space $\mathbb{R}$ to provide an actual example. In this, let $\nu_\mathbb{Q}$ be a standard notation of the rationals. Then we define $\rho_\mathbb{R} :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{R}$ by $\rho_\mathbb{R}(p) = x$ iff $\forall n \in \mathbb{N}\ d(\nu_\mathbb{Q}(p(n)), x) < 2^{-n}$; and consider $\rho_\mathbb{R}$ as the representation of $\mathbb{R}$. We find $< \,\in \mathcal{O}(\mathbb{R} \times \mathbb{R})$ to be computable, thus $\mathbb{R}$ is computably T_2. Moreover, $\mathbb{R}$ satisfies the conditions of Proposition 10.6.

Corollary 10.7. $\overline{\mathbb{R}_\prec} \cong \overline{\mathbb{Q}_\prec}$, $\overline{\mathbb{R}_\succ} \cong \overline{\mathbb{Q}_\succ}$, $\mathbb{R} \cong \mathbb{R}_\prec \wedge \mathbb{R}_\succ$.

Corollary 10.8. $\max : \mathcal{K}(\mathbb{R}) \wedge \mathcal{V}(\mathbb{R}) \to \mathbb{R}$ *is computable.*

Corollary 10.9. $\text{max-value} : (\mathcal{K}(\mathbb{R}) \wedge \mathcal{V}(\mathbb{R})) \times \mathcal{C}(\mathbb{R}, \mathbb{R}) \to \mathbb{R}$ *is computable, where* $\text{max-value}(A, f) = \max\{f(x) \mid x \in A\}$.

Proof. By Proposition 5.5(7) and Proposition 7.4(7), we can compute $f[A] \in (\mathcal{K}(\mathbb{R}) \wedge \mathcal{V}(\mathbb{R}))$, then we use max from Corollary 10.8. □

[14]Here, *dense* is used in the order-theoretic sense, i.e. $\prec$ is dense means that $\forall x, y \in \mathbf{X}\ x \prec y \Rightarrow (\exists z \in \mathbf{X}\ x \prec z \prec y)$.

[15]And here *dense* is used in the topological sense, i.e. means $\forall U \in \mathcal{O}(\mathbf{X})\ U \neq \emptyset \to \exists x \in \mathbf{Y} \cap U$.

Remarks. Corollaries 10.2 and 10.4 are present in Escardó's [26] (2004). The topological counterpart of Corollary 10.4 has been described by Nachbin [40] in 1992. The results on $\mathbb{R}_<$ and $\mathbb{R}_>$ are e.g. included in Weihrauch's [69] (2000).

11. Concluding remarks

It was demonstrated that compactness (Proposition 5.2), T_2 separation (Proposition 6.2), overtness and discreteness (to a lesser extent), and admissibility (Theorem 9.11) all encompass various equivalent properties of represented spaces each characterized by the continuity or computability of specific maps. This marks the extent of possible generalizations of many standard results on computability for sets and functions. In particular, unnecessary restrictions in prior work such as admissibility, second-countability or metrizability are avoided.

As the realizers witnessing the computability of the relevant mappings are generic and independent of the represented spaces involved, the approach presented here is a viable alternative to the introduction of multi-representations of countably based admissibly represented spaces suggested in [56] by Rettinger and Weihrauch.

Moving away from effective topological spaces, i.e. countably based admissible represented spaces, to represented spaces as the primitive objects of investigation for set and function computability opens up options for an integration with the study of hyper-computation. Ziegler [78,79] and Brattka (see [11]) observed that various kinds of hyper-computation such as limit computability or computability with mindchanges can be adequately characterized by means of operators generating new represented spaces from given ones. Based on these, a suitable characterization of functions such as topological closure and interior in terms of computable maps between appropriate represented spaces ought to be possible along the lines of [12].

A more general approach to such *jump operators* can be found in [24] by de Brecht. These operators (which amount to endofunctors) form the basis of a new approach to descriptive set theory [51–53] as suggested by Pauly and de Brecht, which in particular provides derived represented spaces e.g. for the Σ^0_n-measurable functions between given represented spaces, thus generalizing [9] by Brattka. In this, the applicability of descriptive set theory is extended even further than to the Quasi-Polish spaces introduced by de Brecht in [23].

The abstract study of represented spaces seems to hold further aspects to contemplate. Proposition 10.6 already made use of the concept of an effective basis (see also [25]), and a systematic study of those seems to be promising also in order to obtain results on product spaces, e.g. a version of the Tychonoff theorem. Computable measure theory also is amenable to a similar development as computable topology underwent here, as demonstrated by Collins in [19].

Acknowledgements

I would like to thank Vasco Brattka, Martín Escardó and Matthias Schröder for various discussions crucial for the formation of the present work.

The article benefitted from comments by Eike Neumann, Greg Yang and Klaus Weihrauch, as well as the anonymous referees.

References

[1] A. Bauer, Topology and computability, Thesis proposal (for [2]), Carniege Mellon University, 1998.
[2] A. Bauer, The realizability approach to computable analysis and topology, PhD thesis, Carnegie Mellon University, 2000.
[3] A. Bauer, A relationship between equilogical spaces and type two effectivity, *Mathematical Logic Quarterly* **48**(1) (2002), 1–15.
[4] A. Bauer and D. Lesnik, Metric spaces in synthetic topology, *Annals of Pure and Applied Logic* **163**(2) (2012), 87–100.
[5] G. Beer, *Topologies on Closed and Closed Convex Sets*, Kluwer Academic, Dordrecht, 1993.
[6] L. Blum, F. Cucker, M. Shub and S. Smale, *Complexity and Real Computation*, Springer, 1998.

[7] V. Brattka, Recursive characterization of computable real-valued functions and relations, *Theoretical Computer Science* **162** (1996), 45–77.
[8] V. Brattka, Computability on non-separable Banach spaces and Landau's theorem, in: [22], 2005.
[9] V. Brattka, Effective Borel measurability and reducibility of functions, *Mathematical Logic Quarterly* **51**(1) (2005), 19–44.
[10] V. Brattka, Plottable real number functions and the computable graph theorem, *SIAM Journal of Computing* **38** (2008), 303–328.
[11] V. Brattka, M. de Brecht and A. Pauly, Closed choice and a uniform low basis theorem, *Annals of Pure and Applied Logic* **163**(8) (2012), 968–1008.
[12] V. Brattka and G. Gherardi, Borel complexity of topological operations on computable metric spaces, *Journal of Logic and Computation* **19**(1) (2009), 45–76.
[13] V. Brattka and P. Hertling, Continuity and computability of relations, Informatik Berichte 164, FernUniversität Hagen, 1994.
[14] V. Brattka and A. Pauly, Computation with advice, in: *Computability and Complexity in Analysis (CCA 2010)*, Electronic Proceedings in Theoretical Computer Science, Vol. 24, 2010, pp. 41–55.
[15] V. Brattka and G. Presser, Computability on subsets of metric spaces, *Theoretical Computer Science* **305**(1–3) (2003), 43–76.
[16] V. Brattka and K. Weihrauch, Computability on subsets of Euclidean space I: Closed and compact subsets, *Theoretical Computer Science* **219**(1–2) (1999), 65–93.
[17] P. Collins, A computable type theory for control systems, in: *Proc. 48th IEEE Conf. on Decision and Control*, 2009, pp. 5538–5543.
[18] P. Collins, Computable analysis with applications to dynamical systems, Technical Report MAC-1002, CWI Report, Amsterdam, 2010.
[19] P. Collins, Computable stochastic processes, 2014, available at: arXiv:1409.4667.
[20] P. Collins and D.S. Graça, Effective computability of solutions of ordinary differential equations the thousand monkeys approach, *Electronic Notes in Theoretical Computer Science* **221** (2008), 103–114.
[21] P. Collins and D.S. Graça, Effective computability of solutions of differential inclusions: The ten thousand monkeys approach, *Journal of Universal Computer Science* **15**(6) (2009), 1162–1185.
[22] L. Crosilla and P. Schuster, *From Sets and Types to Topology and Analysis: Towards Practicable Foundations for Constructive Mathematics*, Oxford Logic Guides, Vol. 48, Clarendon Press, Oxford, 2005.
[23] M. de Brecht, Quasi-Polish spaces, *Annals of Pure and Applied Logic* **164**(3) (2013), 354–381.
[24] M. de Brecht, Levels of discontinuity, limit-computability, and jump operators, in: *Logic, Computation, Hierarchies*, V. Brattka, H. Diener and D. Spreen, eds, de Gruyter, 2014, pp. 79–108, available at: arXiv:1312.0697.
[25] M. de Brecht, M. Schröder and V. Selivanov, Base-complexity classifications of QCB_0-spaces, in: *Evolving Computability*, A. Beckmann, V. Mitrana and M. Soskova, eds, Lecture Notes in Computer Science, Vol. 9136, Springer, 2015, pp. 156–166.
[26] M. Escardó, Synthetic topology of datatypes and classical spaces, *Electronic Notes in Theoretical Computer Science* **87** (2004), 21–156.
[27] M. Escardó, Infinite sets that admit fast exhaustive search, in: *Proc. 22nd Annual IEEE Symposium on Logic in Computer Science*, IEEE Computer Society Press, 2007, pp. 443–452.
[28] M. Escardó, Exhaustible sets in higher-type computation, *Logical Methods in Computer Science* **4**(3) (2008), 3.
[29] M. Escardó, Intersections of compactly many open sets are open, 2009.
[30] J.M.G. Fell, A Hausdorff topology for the closed subsets of a locally compact non-Hausdorff space, *Proceedings of the American Mathematical Society* **13** (1962), 472–476.
[31] S. Galatolo, M. Hoyrup and C. Rojas, Dynamics and abstract computability: computing invariant measures, *Discrete and Continuous Dynamical Systems* **29**(1) (2011), 193–212.
[32] G. Giertz, K.H. Hofmann, K. Keimel, J.D. Lawson, M.W. Mislove and D.S. Scott, *A Compendium of Continuous Lattices*, Springer, 1980.

[33] Z. Iljazović, Chainable and circularly chainable co-c.e. sets in computable metric spaces, *Journal of Universal Computer Science* **15**(6) (2009), 1206–1235.

[34] Z. Iljazović, Co-c.e. spheres and cells in computable metric spaces, *Logical Methods in Computer Science* **7**(3) (2011), 1–21.

[35] B.M. Kapron, Feasibly continuous type-two functionals, *Computational Complexity* **8**(2) (1999), 188–201.

[36] A. Kawamura and S. Cook, Complexity theory for operators in analysis, *ACM Transactions on Computation Theory* **4**(2) (2012), 5.

[37] T. Kihara and A. Pauly, Point degree spectra of represented spaces, 2014, available at: arXiv:1405.6866.

[38] C. Kreitz and K. Weihrauch, Theory of representations, *Theoretical Computer Science* **38** (1985), 35–53.

[39] S. Le Roux and A. Pauly, Finite choice, convex choice and finding roots, *Logical Methods in Computer Science* **11**(4) (2015), 6.

[40] N. Leopoldo, Compact unions of closed subsets are closed and compact intersections of open subsets are open, *Portugaliae Mathematica* **49**(4) (1992), 403–409.

[41] Z. Li and J.D. Hamkins, On effectiveness of operations on countable ordinals, Unpublished notes.

[42] P. Lietz, From constructive mathematics to computable analysis via the realizability interpretation, PhD thesis, Technische Universität Darmstadt, 2004.

[43] J. Miller, Effectiveness for embedded spheres and balls, *Electronic Notes in Theoretical Computer Science* **66** (2002), 127–138.

[44] J.S. Miller, Π^0_1 classes in computable analysis and topology, PhD thesis, Cornell University, 2002.

[45] A. Pauly, Representing measurement results, *Journal of Universal Computer Science* **15**(6) (2009), 1280–1300.

[46] A. Pauly, Computable metamathematics and its application to game theory, PhD thesis, University of Cambridge, 2012.

[47] A. Pauly, Multi-valued functions in computability theory, in: *How the World Computes*, S. Cooper, A. Dawar and B. Löwe, eds, Lecture Notes in Computer Science, Vol. 7318, Springer, 2012, pp. 571–580.

[48] A. Pauly, Computability on the countable ordinals and the Hausdorff–Kuratowski theorem, 2015, available at: arXiv:1501.00386.

[49] A. Pauly, Computability on the countable ordinals and the Hausdorff–Kuratowski theorem (extended abstract), in: *Mathematical Foundations of Computer Science 2015*, G.F. Italiano, G. Pighizzini and D.T. Sannella, eds, Lecture Notes in Computer Science, Vol. 9234, Springer, 2015, pp. 407–418.

[50] A. Pauly, Many-one reductions and the category of multivalued functions, *Mathematical Structures in Computer Science* (2015), doi:10.1017/S0960129515000262, available at: arXiv:1102.3151.

[51] A. Pauly and M. de Brecht, Non-deterministic computation and the Jayne–Rogers theorem, in: *Developments in Computational Models 2012 (DCM 2012)*, Electronic Proceedings in Theoretical Computer Science, Vol. 143, 2014, pp. 87–96.

[52] A. Pauly and M. de Brecht, Descriptive set theory in the category of represented spaces, in: *30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2015, pp. 438–449.

[53] A. Pauly and M. de Brecht, Towards synthetic descriptive set theory: An instantiation with represented spaces, available at: arXiv:1307.1850.

[54] A. Pauly and M. Ziegler, Relative computability and uniform continuity of relations, *Journal of Logic & Analysis* **5** (2013), 7.

[55] R. Rettinger, Compactness and the effectivity of uniformization, in: *How the World Computes*, S. Barry Cooper, A. Dawar and B. Löwe, eds, Lecture Notes in Computer Science, Vol. 7318, Springer, 2012, pp. 616–625.

[56] R. Rettinger and K. Weihrauch, Products of effective topological spaces and a uniformly computable Tychonoff theorem, *Logical Methods in Computer Science* **9**(4) (2013), 14.

[57] M. Schröder, Admissible representations for continuous computations, PhD thesis, FernUniversität Hagen, 2002.

[58] M. Schröder, Effectivity in spaces with admissible multirepresentations, *Mathematical Logic Quarterly* **48**(1) (2002), 78–90.

[59] M. Schröder, Extended admissibility, *Theoretical Computer Science* **284**(2) (2002), 519–538.

[60] M. Schröder, Spaces allowing type-2 complexity theory revisited, *Mathematical Logic Quarterly* **50**(4–5) (2004), 443–459.
[61] M. Schröder, Admissible representations for probability measures, *Mathematical Logic Quarterly* **53**(4) (2007), 431–445.
[62] M. Schröder, A Hofmann–Mislove theorem for Scott open sets, 2015, available at: arXiv:1501.06452.
[63] P. Taylor, A lambda calculus for real analysis, *Journal of Logic & Analysis* **2**(5) (2010), 1–115.
[64] P. Taylor, Foundations for computable topology, in: *Foundational Theories of Classical and Constructive Mathematics*, The Western Ontario Series in Philosophy of Science, Vol. 76, Springer, 2011, pp. 265–310.
[65] L. Vietoris, Bereiche zweiter Ordnung, *Monatsheft für Mathematik* **32** (1922), 258–280.
[66] K. Weihrauch, Type 2 recursion theory, *Theoretical Computer Science* **38** (1985), 17–33.
[67] K. Weihrauch, *Computability*, Monographs on Theoretical Computer Science, Springer-Verlag, 1987.
[68] K. Weihrauch, Computability on the probability measures on the Borel sets of the unit interval, *Theoretical Computer Science* **219** (1999), 421–437.
[69] K. Weihrauch, *Computable Analysis*, Springer-Verlag, 2000.
[70] K. Weihrauch, Computational complexity on computable metric spaces, *Mathematical Logic Quarterly* **49**(1) (2003), 3–21.
[71] K. Weihrauch, The computable multi-functions on multi-represented sets are closed under programming, *Journal of Universal Computer Science* **14**(6) (2008), 801–844.
[72] K. Weihrauch, Computable separation in topology, from T_0 to T_2, *Journal of Universal Computer Science* **16**(18) (2010), 2733–2753.
[73] K. Weihrauch, Computably regular topological spaces, *Logical Methods in Computer Science* **9**(4–5) (2013).
[74] K. Weihrauch and T. Grubba, Elementary computable topology, *Journal of Universal Computer Science* **15**(6) (2009), 1381–1422.
[75] K. Weihrauch and G. Schäfer, Admissible representations of effective CPO's, *Theoretical Computer Science* **26**(1–2) (1983), 131–147.
[76] M. Ziegler, Computable operators on regular sets, *Mathematical Logic Quarterly* **50** (2004), 392–404.
[77] M. Ziegler, Computability and continuity on the real arithmetic hierarchy and the power of type-2 nondeterminism, in: *Proc. CiE 2005*, B.S. Cooper, B. Löwe and L. Torenvliet, eds, Lecture Notes in Computer Science, Vol. 3526, Springer, 2005, pp. 562–571.
[78] M. Ziegler, Real hypercomputation and continuity, *Theory of Computing Systems* **41** (2007), 177–206.
[79] M. Ziegler, Revising type-2 computation and degrees of discontinuity, *Electronic Notes in Theoretical Computer Science* **167** (2007), 255–274.