Computability of continuous solutions of higher-type equations

Martín Escardó
School of Computer Science, University of Birmingham, UK

Work in progress, draft version of July 25, 2008

1 Introduction

We investigate the following problem in higher-type computation:

> Given computable functionals $f \colon X \to Y$ and $y \in X$, compute $x \in X$ such that $f(x) = y$, if such an x exists.

We show that if x is unique and X and Y are retracts of Kleene–Kreisel spaces with X exhaustible in the sense of [5], then x is computable uniformly in f, y and $\forall_X$. The computation of unique solutions of equations of the form $g(x) = h(x)$ with $g, h \colon X \to Y$ is easily reduced to the previous case, because there are group structures on the ground types that can be lifted componentwise to product types and pointwise to function types. And, by cartesian closedness, the case in which g and h computably depend on parameters, and in which the solution computably depends on the same parameters, is covered. Moreover, because Kleene–Kreisel spaces are closed under finite products and countable powers, this includes the solution of finite and countably infinite systems of equations with functionals of finitely many or countably infinitely many variables.

We also consider generalizations to computational metric spaces that apply to computational analysis, where f can be a functional and x a function. And we develop examples of sets of analytic functions that are exhaustible and can play the role of the space X.

The nature of the problem forces us to consider partial functionals defined on total data, because the unique-solution functional is promised to be defined on (f, y, ϕ) iff there is a unique x with $f(x) = y$ and $\phi = \forall_X$, and cannot fulfill a better promise (see [4] and the discussion in Section 3). Similar headaches involving totality and partiality took place in our previous work on exhaustible sets. Before developing the above results, we consider a remedy in Section 4. In Section 5 we reformulate the definitions and results of our previous work on exhaustible sets in a technically and conceptually simpler way using the set-up of Section 4. And then finally from Section 6 onwards we proceed to develop the results discussed above.

2 Prerequisites and disclaimer

For the moment, we assume the notation and terminology of our paper paper "Exhaustible sets in higher-type computation" [5]. At present, the exposition lacks proper background and credits and citations, and is written for readers who are conversant with a particular, but also general, approach to computability. Later we'll also address a more general set of readers. This is a preliminary report, with emphasis on the technical results. A more detailed version will be written later.

3 Discussion on uniqueness and exhaustibility

This section justifies the assumptions made in later sections, and can be safely skipped. Given Kleene–Kreisel spaces X and Y, a continuous function $f\colon X \to Y$ and $y \in Y$, we wish to compute $x \in X$, uniformly in f and y, such that

$$f(x) = y.$$

We discuss several cases for X and Y and explain why, in general, further assumptions and data are required.

The simplest case is $X = Y = \mathbb{N}$, for which the algorithm

$$\mu x. f(x) = y$$

computes a solution iff a solution exists. This is subsumed by the next case.

Consider X arbitrary and $Y = \mathbb{N}$. By the Kleene–Kreisel density theorem, X has a computable dense sequence $\delta\colon \mathbb{N} \to X$, and, by continuity of f and discreteness of $\mathbb{N}$, if the equation has a solution, there is one of the form $x = \delta_n$ for some n. Hence the algorithm

$$x = \delta_{\mu n. f(\delta_n) = y}$$

computes a solution iff a solution exists. Moreover, in this particular case it is semi-decidable whether a solution exists, with the algorithm $\exists n. f(\delta_n) = y$.

Now consider $X = 2$ and $Y = \mathbb{N}^{\mathbb{N}}$. Then a function $f\colon X \to Y$ amounts to two functions $f_0, f_1\colon \mathbb{N} \to \mathbb{N}$, and computing a solution to the above equation amounts to finding $i \in 2$ such that $f_i = y$, that is, $f_i(n) = y(n)$ for all $n \in \mathbb{N}$. In other words, under the assumption that

> $f_0 = y$ or $f_1 = y$,

we want to find i such that $f_i = y$. If the only data supplied to the desired algorithm are f_0, f_1, y, this is not possible, because no finite amount of information about the data can determine that one particular disjunct holds (a similar situation is worked out in detail below). However, if we instead assume that

> one of $f_0 = y$ and $f_1 = y$ holds, but not both,

then we can compute i as follows:

> Find the least n such that $f_0(n) \neq y(n)$ or $f_1(n) \neq y(n)$, and let i be the unique number such that $f_i(n) = y(n)$.

Thus, in general, it is not possible to compute solutions unless we know that they are unique, and in this particular case one can compute unique solutions. This kind of phenomenon is well known — see e.g. [4].

Next consider $X = \mathbb{N}$ and $Y = \mathbb{N}^{\mathbb{N}}$, and assume that the equation $f(x) = y$ has a unique solution. Now it is no longer possible to compute it uniformly in f and y. For suppose there existed a computable partial functional $s\colon (\mathbb{N}^{\mathbb{N}})^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$, defined on some superset of

$$S = \{(f, y) \mid f(x) = y \text{ has a unique solution } x_0\},$$

such that $x_0 = s(f, y)$ is the solution for $f, y \in S$. By continuity, for any $(f, y) \in S$ there is a number n such that $s(f, y) = x_0 = s(g, y)$ for every g such that $g(x)(i) = f(x)(i)$ for all $x < n$ and $i < n$. W.l.o.g. we can assume that $x_0 < n$ by replacing n by $\max(n, x_0) + 1$ if necessary. Choose g defined by

$$g(x)(i) = \begin{cases} f(x)(i) & \text{if } x < n \text{ and } i < n, \\ y(i) & \text{if } x = n, \\ y(i) + 1 & \text{otherwise.} \end{cases}$$

By construction, $x = n$ is the unique solution of $g(x) = y$, and hence $s(g, y) = n$, which contradicts $s(g, y) = x_0$, and concludes the proof of the impossibility claim. However, if we know that there is a unique solution in a finite set $K \subseteq \mathbb{N}$, then the solution can be found uniformly in f, y and a finite enumeration $e_0, \dots, e_{k-1}$ of K, as follows:

> Find n and $j < k$ such that the decidable conditions $\forall i < n.f(e_j)(i) = y(i)$ and $\forall l < k, l \neq j.\exists i < n.f(e_l)(i) \neq y(i)$ hold, and take $x = e_j$.

This generalizes the situation $X = 2$, and is a particular case of Theorem 6.1 below, which shows that unique solutions in exhaustible subsets X of Kleene–Kreisel spaces are computable uniformly in f, y and $\forall_X$.

4 Computational spaces

In order to simplify the development, both technically and conceptually, we work with a computational category investigated by Bauer [1], with continuous versions discussed by Bauer, Birkedal and Scott [3]. This goes back to a 1996 unpublished manuscript by Scott in which he introduced equilogical spaces. In summary, we work with the category $\mathrm{Mod}(\mathrm{EffScott})$, where $\mathrm{EffScott}$ is the cartesian closed category of computable maps of effectively given Scott domains. We propose to refer to the objects of this category as *computational spaces* and to its morphisms as *computable functions*. This general terminology for the objects is justified by the fact that this category incorporates, as discussed in [3, 2], all known models of effective computation, including effectively given domains, Kleene–Kreisel spaces, Berger's domains with totality, Weihrauch's TTE, Blanck's domain representations, Schröder and Simpson's QCB spaces. But this is only a justification for our terminology: our main reason for considering this category is simplicity in the formulation and proofs of theorems, rather than just generality.

4.1 Definition of the category of computational spaces

Explicitly, the category we are interested in is defined as follows:

1. A *computational space*, or *space* for short, is a list $X = (|X|, D_X, T_X, \rho_X)$ consisting of an underlying set $|X|$, often denoted by X by an abuse of notation, equipped with an effectively given domain D_X together with a set $T_X \subseteq D_X$ and a surjection $\rho_X : T_X \to |X|$.

   The structure of the underlying set X is presented in diagrammatic form as

   $$D_X \hookleftarrow T_X \xrightarrow{\rho_X} X.$$

   The idea is that the domain D_X collects all possible total and partial data that a particular computational mechanism can produce, the subset T_X collects the data one is interested in for a particular purpose, and the set X identifies different pieces of relevant data that one regards as equivalent for that purpose.

2. A computable map $f : X \to Y$ is a function of the underlying sets that is tracked by some computable map $\ulcorner f \urcorner : D_X \to D_Y$, in the sense that the diagram

   $$\begin{tikzcd} D_X \arrow[d, "\ulcorner f \urcorner"'] & T_X \arrow[l, hook'] \arrow[r, two heads, "\rho_X"] \arrow[d, dashed] & X \arrow[d, "f"] \\ D_Y & T_Y \arrow[l, hook'] \arrow[r, two heads, "\rho_Y"] & Y \end{tikzcd}$$

commutes for some function $T_X \to T_Y$, which then must be the restriction of $\ulcorner f \urcorner$. This is equivalent to saying that, for any $d \in T_X$, we have that $\ulcorner f \urcorner(d) \in T_Y$ and

$$f(\rho_X(d)) = \rho_Y(\ulcorner f \urcorner(d)).$$

3. Identities and composition are inherited from the category of sets.

Notice that, in this category, computable maps $f\colon X \to Y$ are always defined, in the mathematical sense, for every $x \in X$. Partiality, in the computational sense, if desired, is achieved by allowing Y to have undefined or partially defined elements. We denote finite products and exponentials in this category by the usual notations $X \times Y$ and Y^X, and sometimes $(X \to Y)$. For their construction, see [1, 3]. We just emphasize that the underlying set of Y^X consists of the maps that are tracked by continuous, rather than computable functions, despite the fact that morphisms of the category are tracked by computable functions. Computable maps, and elements of exponentials, are continuous with respect to the computational topology:

Definition 4.1. We endow a computational space X with the ρ_X-quotient topology of the relative Scott topology of D_X on T_X, called the *computational topology of* X. □

Examples 4.2. We agree that

1. An effectively given domain D is regarded as the space described by

$$D = D = D.$$

Then a computable map $D \to E$ of effectively given domains is a computable map in the usual sense. That is, this construction is a full embedding of the category of effectively given domains into the category of computational spaces. Moreover, this embedding preserves the cartesian closed structure.

2. This applies, in particular, to the domains D_σ of the simple type hierarchy.

3. A Kleene–Kreisel space C_σ is regarded as the space described by

$$D_\sigma \hookleftarrow T_\sigma \xrightarrow{\rho_\sigma} C_\sigma.$$

Then again a computable map $C_\sigma \to C_\tau$ is a computable map in the usual sense, and this construction is a full embedding of the category of Kleene–Kreisel spaces with computable maps into the category of computational spaces, with the embedding preserving the cartesian closed structure [3].

4. $\mathbb{N}$ denotes the Kleene–Kreisel space C_ι,

$$\mathbb{N}_\perp \hookleftarrow \mathbb{N} = \mathbb{N},$$

which has the discrete topology.

5. 2 denotes the Kleene–Kreisel space C_o, with underlying set $\{0, 1\}$,

$$2_\perp \hookleftarrow 2 = 2,$$

which again has the discrete topology (rather than the Sierpinski topology).

NB. Of course, it is not necessary to include the base type o because 2 also arises as a computable retract of the Kleene–Kreisel space $\mathbb{N}$. □

4.2 Subspaces, images and quotients

We'll often use the following constructions to build spaces:

Definition 4.3. A subspace inclusion $S \to X$ is a computable map of the form:

$$\begin{tikzcd} D_S \arrow[d, equal] & T_S \arrow[l, hook'] \arrow[r, two heads, "\rho_S"] \arrow[d, hook] & S \arrow[d, hook] \\ D_X & T_X \arrow[l, hook'] \arrow[r, two heads, "\rho_X"] & X. \end{tikzcd}$$

Given a space X and a subset S, the relative subspace structure on S is given by taking $D_S = D_X, T_S = \rho_X^{-1}(S)$ and ρ_S the restriction of ρ_X. $\square$

Definition 4.4. The image of a computable map $f\colon X \to Y$ is the set-theoretical image $f[X]$ equipped with the subspace structure. $\square$

Definition 4.5. A quotient map $q\colon A \to X$ is a computable map of the form

$$\begin{tikzcd} D_A \arrow[d, equal] & T_A \arrow[l, hook'] \arrow[r, two heads, "\rho_A"] \arrow[d, equal] & A \arrow[d, two heads, "q"] \\ D_X & T_X \arrow[l, hook'] \arrow[r, two heads, "\rho_X"] & X. \end{tikzcd}$$

Given a space A, a set X, and a surjection $q\colon A \to X$, the quotient structure on X is then given by $D_X = D_A, T_X = T_A$, and $\rho_X = q \circ \rho_A$. $\square$

The following will play an important role:

Examples 4.6.

1. The sets $\mathbb{Z}$ of integers and $\mathbb{Q}$ of rational numbers are regarded as discrete spaces via standard codings based on natural numbers.
2. Then $\mathbb{Q}^{\mathbb{N}}$ is also a space by cartesian closedness, and so is its subset $\mathrm{Cauchy}(\mathbb{Q})$ of Cauchy sequences $q \in \mathbb{Q}^N$ such that $|q_n - q_{n+1}| < 2^{-n}$ with the subspace structure.
3. The *real line* in the category of computational spaces is the set $\mathbb{R}$ of real numbers with the quotient structure induced by the surjection $\lim\colon \mathrm{Cauchy}(\mathbb{Q}) \to \mathbb{R}$ that takes a sequence to its limit.
4. We endow $[0, \infty)$, $[0, 1]$, $[-1, 1]$ etc. with the relative subspace structure of $\mathbb{R}$. $\square$

This material can and should be put in a proper categorical setting, but at the moment we are giving priority to applications.

4.3 Representing spaces

For certain applications, we consider the structure T_X of a space X as a space on its own (an object of the category, denoted by $\ulcorner X \urcorner$), and the structure $\rho_X : T_X \to X$ of X as a computable map (a morphism of the category, denoted by the same name without danger of ambiguity):

Definition 4.7. For any space X, define the *representing space* $\ulcorner X \urcorner$ by

$$D_{\ulcorner X \urcorner} = D_X, \quad T_{\ulcorner X \urcorner} = |\ulcorner X \urcorner| = T_X, \quad \rho_{\ulcorner X \urcorner} = \mathrm{id}_{T_X}\,.$$

In diagrammatic form, $\ulcorner X \urcorner$ is the space

$$\begin{tikzcd} D_X & T_X \arrow[l, hook'] \arrow[r, equal] & T_X, \end{tikzcd}$$

which of course is also written as

$$D_{\ulcorner X \urcorner} \hookleftarrow T_{\ulcorner X \urcorner} \xrightarrow{\rho_{\ulcorner X \urcorner}} \ulcorner X \urcorner,$$

using our general convention for denoting spaces. Then ρ_X is regarded as a quotient map $\ulcorner X \urcorner \to X$, called the *representation function*, as follows:

$$\begin{tikzcd} D_X \arrow[d, equal] & T_X \arrow[l, hook'] \arrow[r, equal] \arrow[d, equal] & T_X \arrow[d, two heads, "\rho_X"] \\ D_X & T_X \arrow[l, hook'] \arrow[r, two heads, "\rho_X"] & X. \end{tikzcd}$$

Because it is tracked by the identity, it is computable. □

Notice that $\ulcorner X \times Y \urcorner = \ulcorner X \urcorner \times \ulcorner Y \urcorner$. Of course, the map $\ulcorner f \urcorner$ defined below is not uniquely determined by f, and hence there is no computable map $\ulcorner - \urcorner \colon Y^X \to \ulcorner Y \urcorner^{\ulcorner X \urcorner}$.

Lemma 4.8. *For any computable function $f \colon X \to Y$ there is a computable function $\ulcorner f \urcorner \colon \ulcorner X \urcorner \to \ulcorner Y \urcorner$ such that $f \circ \rho_X = \rho_Y \circ \ulcorner f \urcorner$:*

$$\begin{tikzcd} \ulcorner X \urcorner \arrow[r, "\ulcorner f \urcorner"] \arrow[d, two heads, "\rho_X"'] & \ulcorner Y \urcorner \arrow[d, two heads, "\rho_Y"] \\ X \arrow[r, "f"] & Y. \end{tikzcd}$$

Moreover, if $f \colon X \to Y$ is a surjection, then so is any representative $\ulcorner f \urcorner$.

Proof. Take the restriction of some computable $\ulcorner f \urcorner \colon D_X \to D_Y$ that tracks $f \colon X \to Y$ to $T_X \to T_Y$. This regarded as a map $\ulcorner f \urcorner \colon \ulcorner X \urcorner \to \ulcorner Y \urcorner$ is computable, because it is also tracked by $\ulcorner f \urcorner \colon D_X \to D_Y$. If f is a surjection, three arrows of the square, without counting $\ulcorner f \urcorner$, are surjections, and hence the remaining one must also be a surjection. □

5 Exhaustible spaces

In previous work we investigated exhaustible *subsets* of effectively given domains, with emphasis on exhaustible sets of total elements, or more precisely entire sets [5]. Here we define exhaustible *spaces* and transfer results for them from that work.

5.1 Exhaustibility and related notions

Definition 5.1.

1. A space K is called *exhaustible* if the universal quantification functional

$$\forall_K \colon 2^K \to 2$$

defined by

$$\forall_K(p) = 1 \iff p(x) = 1 \text{ for all } x \in K$$

is computable.

2. It is called *searchable* if there is a computable selection functional

$$\varepsilon_K \colon 2^K \to K$$

such that for all $p \in 2^K$, if there is $x \in K$ with $p(x) = 1$ then $x = \varepsilon_K(p)$ is an example.

3. A set $F \subseteq X$ is *decidable* if its characteristic map $X \to 2$ is computable. □

Equivalently, K is exhaustible iff the map $\exists_K : 2^K \to 2$ defined by

$$\exists_K(p) = 1 \iff p(x) = 1 \text{ for some } x \in K$$

is computable, as $\exists_K$ and $\forall_K$ are inter-definable using the De Morgan laws. If K is searchable, then it is exhaustible, because

$$\exists_K(p) = p(\varepsilon_K(p)).$$

The empty set is exhaustible, but not searchable, because there is no map $2^{\emptyset} \to \emptyset$.

We regard a subset K of a domain D as a space by firstly regarding D as a space (Examples 4.2) and then endowing K with subspace structure (Definition 4.3).

Lemma 5.2. *The following are equivalent for any subset K of a domain D:*

1. *K is exhaustible as a set, in the sense of [5].*
2. *K is exhaustible as a space, in the sense of Definition 5.1.*

The two statements remain equivalent with searchable *in place of* exhaustible.

Proof. We just observe that if a map $\ulcorner p \urcorner \colon D \to 2_\perp$ tracks a map $p\colon K \to 2$, then $\ulcorner p \urcorner$ is defined on the set K, in the terminology of [5]. □

5.2 Exhaustible subspaces of Kleene–Kreisel spaces

Lemma 5.3.

1. *The Cantor space $2^{\mathbb{N}}$ is searchable.*
2. *Any exhaustible subspace of a Kleene–Kreisel space is compact in the computational topology, and moreover, if it is non-empty, it is searchable, a computable retract of the Kleene–Kreisel space, and a computable image of the Cantor space.*
3. *Searchable spaces are closed under computable images, finite intersections with decidable sets, and finite products.*
4. *Searchable spaces that share the same domain structure are closed under countable products.*

Proof. (1): It is searchable as a subset of a domain, using Berger's algorithm [5]. (2): This is formulated and proved for entire sets in [5]. (3): The algorithms given in [5] apply more generally to computational spaces. (4): Use the product algorithm of [5] to building the tracking map. □

It is well known that any Kleene–Kreisel space is a computable retract of a Kleene–Kreisel space of the form $\mathbb{N}^X$, and we use this in our arguments of Section 6 below, which then automatically apply to kk-spaces:

Definition 5.4. A *kk-space* is a computable retract of a Kleene–Kreisel space. □

Because retracts compose, the kk-spaces are precisely the computable retracts of the Kleene–Kreisel spaces of the form $\mathbb{N}^X$. By a general and easy argument, kk-spaces are closed under the formation of finite products and exponentials. Moreover, they form the smallest collection of spaces containing the Kleene–Kreisel spaces and satisfying this condition. For example, the discrete space $3 = \{-1, 0, 1\}$ and the exponential $3^{\mathbb{N}}$ are kk-spaces but not Kleene–Kreisel spaces.

Lemma 5.5.

1. *Any exhaustible kk-space is compact, and searchable if it is non-empty.*

2. *An exhaustible space is a kk-space iff it is a subspace of some Kleene–Kreisel space.*

Proof. (1): Because kk-spaces are subspaces of Kleene–Kreisel spaces. (2): Because any non-empty exhaustible subspace of a Kleene–Kreisel space is a computable retract. □

5.3 Spaces with exhaustible kk-spaces of representatives

The remainder of this section is not needed until Section 7. We observed in [5] that if a space X is connected, as will be the case in applications of Theorem 7.1 to analysis, computable maps $p\colon X \to 2$ are constant, and hence X is trivially exhaustible if it has some computable point x_0, as its quantifier is computable as $\forall_X(p) = p(x_0)$. On the other hand, any kk-space X is totally separated (the clopen sets, or equivalently, the continuous predicates $p\colon X \to 2$, separate the points), which implies that it is totally disconnected (the connected components are singletons). This partly explains why kk-spaces have the good properties discussed above, for example that exhaustibility implies compactness, which, as we have just seen, fails for connected spaces. Moreover, kk-spaces are computationally totally separated, in the sense that the computable predicates separate the points. This motivates the use, in Theorem 7.1, of computational (metric) spaces X with exhaustible kk-spaces of representatives. Such a space, being the computable (and hence continuous) image of the representing map, is exhaustible and compact.

In general, kk-spaces are topological *quotients* of subspaces of hereditarily total elements of domains under the Scott topology, as is well known. By the continuous versions of the constructions of [5], every *compact* kk-space arises as a topological *subspace* of total elements of such a domain. Concrete examples are given below. Recall the notion of representing space given in Definition 4.7.

Definition 5.6. A computational space is called *kk-exhaustible* if it is isomorphic to some space X such that its representing space $\ulcorner X \urcorner$ is an exhaustible kk-space. □

(The isomorphism is automatically required to be computable, because the morphisms of our underlying category are computable maps.)

Examples 5.7. The following are kk-exhaustible:

1. *Any interval $[a, b]$ with $a, b \in \mathbb{R}$ computable.*

   Given our particular construction of $[-1, 1]$ in Examples 4.6, its representing space is not exhaustible. But, as is well known $[-1, 1]$ is isomorphic to the set $[-1, 1]$ endowed with signed-digit binary representation, with representing space $3^{\mathbb{N}}$ where $3 = \{-1, 0, 1\}$, which is an exhaustible kk-space. The general case $[a, b]$ is easily reduced to this.

2. *Finite products, countable powers and retracts of kk-exhaustible spaces.*

   We omit the routine details, at least for the moment.

Of course, the real line is not kk-exhaustible, because it is not compact. □

We'll see in Section 8 that interesting subspaces of analytic functions of the space $\mathbb{R}^{[-\epsilon,\epsilon]}$, with $0 < \epsilon < 1$, are kk-exhaustible, which will allow us to compute solutions of functional equations with analytic unknowns.

6 Equations over Kleene–Kreisel spaces

Theorem 6.1. *If $f\colon X \to Y$ is a computable map of kk-spaces with X exhaustible, and $y \in Y$ is computable, then, uniformly in $\forall_X$, f, and y:*

1. *It is semi-decidable whether the equation $f(x) = y$ fails to have a solution $x \in X$.*
2. *If $f(x) = y$ has a unique solution $x \in X$, then it is computable.*

Hence if $f\colon X \to Y$ is a computable bijection then it has a computable inverse, uniformly in $\forall_X$ and f.

The conclusion is a computational version for kk-spaces of the topological theorem that any continuous bijection of compact Hausdorff spaces is a homeomorphism. This gives an alternative route to the following fact established in [5]:

Theorem 6.2. *Any exhaustible kk-space is computably homeomorphic to an exhaustible subspace of the Baire space $\mathbb{N}^{\mathbb{N}}$.*

Proof. Let K be an exhaustible kk-space, let $s\colon K \to \mathbb{N}^Z$ and $r\colon \mathbb{N}^Z \to K$ be computable maps with $r \circ s = \mathrm{id}_K$ and Z a Kleene–Kreisel space, and let $\delta_n \in Z$ be a computable dense sequence. The subspace $X = s(K) \subseteq \mathbb{N}^Z$, being a computable image of an exhaustible space, is itself exhaustible. As in [5], we consider the map $X \to \mathbb{N}^{\mathbb{N}}$ that sends $u \in X$ to the sequence $u(\delta_n)$, but we argue using Theorem 6.1 instead. Let $f\colon X \to Y$ be the restriction of this map to its image $Y \subseteq \mathbb{N}^{\mathbb{N}}$. By density, f is one-to-one, and, by construction, it is onto, and hence it has a computable inverse. Therefore there is computable map $g\colon K \to Y$ defined by $g(k) = f(s(k))$ with computable inverse given by $g^{-1}(\alpha) = r(f^{-1}(\alpha))$. □

We now prove Theorem 6.1. The following will be applied to semi-decide that equations fail to have solutions:

Lemma 6.3. *Let X be an exhaustible kk-space and $K_n \subseteq X$ be a sequence of sets that are decidable uniformly in n and satisfy $K_n \supseteq K_{n+1}$.*

Emptiness of $\bigcap_n K_n$ is semi-decidable,

uniformly in the quantifier of X and the sequence of decision procedures for K_n.

Proof. Because X is compact by exhaustibility, K_n is also compact as it is closed. Because X is Hausdorff, $\bigcap_n K_n = \emptyset$ iff there is n such that $K_n = \emptyset$. But emptiness of this set is decidable uniformly in n by the algorithm $\forall x \in X. x \notin K_n$. Hence a semi-decision procedure is given by $\exists n. \forall x \in X. x \notin K_n$. □

As a preparation for a lemma that will be applied to compute unique solutions, notice that if a singleton $\{u\} \subseteq \mathbb{N}^Z$ is exhaustible, then the function u is computable, because $u(z) = \mu m. \forall v \in \{u\}. u(z) = m$. Moreover, u is computable uniformly in $\forall_{\{u\}}$, in the sense that there is a computable functional

$$U\colon S \to \mathbb{N}^Z \quad \text{with} \quad S = \{\phi \in 2^{2^{\mathbb{N}^Z}} \mid \phi = \forall_{\{v\}} \text{ for some } v \in \mathbb{N}^Z\},$$

such that

$$u = U\left(\forall_{\{u\}}\right),$$

namely

$$U(\phi)(z) = \mu m. \phi(\lambda u. u(z) = m).$$

Lemma 6.4 below generalizes this, using an argument from [5] that was originally used to prove that non-empty exhaustible subsets of kk-spaces are computable images of the Cantor space and hence searchable. Here we find additional applications and further useful generalizations.

Lemma 6.4. *Let X be a kk-space and $K_n \subseteq X$ be a sequence of sets that are exhaustible uniformly in n and satisfy $K_n \supseteq K_{n+1}$.*

If $\bigcap_n K_n$ is a singleton $\{x\}$, then x is computable,

uniformly in the sequence $\forall_{K_n}$.

Proof. Let X be a kk-space and $s\colon X \to \mathbb{N}^Z$ and $r\colon \mathbb{N}^Z \to X$ be computable functions with $r \circ s = \mathrm{id}_X$. It suffices to show that the function $u = s(x) \in \mathbb{N}^Z$ is computable, because $x = r(u)$. The sets $L_n = s(K_n) \subseteq \mathbb{N}^Z$, being computable images of exhaustible sets, are themselves exhaustible. For any $z \in Z$, the set $U_z = \{v \in \mathbb{N}^Z \mid v(z) = u(z)\}$ is clopen and $\bigcap_n L_n = \{u\} \subseteq U_z$. Because $\mathbb{N}^Z$ is Hausdorff, because $L_n \supseteq L_{n+1}$, because each L_n is compact and because U_z is open, there is n such that $L_n \subseteq U_z$, that is, $v \in L_n$ implies $v(z) = u(z)$. Therefore, for every $z \in Z$ there is n such that $v(z) = w(z)$ for all $v, w \in L_n$. Now, the function $n(z) = \mu n.\forall v, w \in L_n.v(z) = w(z)$ is computable by the exhaustibility of L_n. But $u \in L_{n(z)}$ for any $z \in Z$ and therefore u is computable by exhaustibility as $u(z) = \mu m.\forall v \in L_{n(z)}.v(z) = m$. □

To build sets K_n suitable for applying this lemma, we use:

Lemma 6.5. *For every kk-space X there is a family $(=_n)$ of equivalence relations that are decidable uniformly in n and satisfy*

$$\begin{aligned} x = x' \quad &\iff \quad \forall n.\, x =_n x', \\ x =_{n+1} x' \quad &\implies \quad x =_n x'. \end{aligned}$$

Proof. Let X be a Kleene–Kreisel space and $s\colon X \to \mathbb{N}^Z$ and $r\colon \mathbb{N}^Z \to X$ be computable maps with $r \circ s = \mathrm{id}_X$. By the density theorem, there is a computable dense sequence $\delta_n \in Z$. Then the definition

$$x =_n x' \iff \forall i < n.s(x)(\delta_i) = s(x')(\delta_i)$$

clearly produces an equivalence relation that is decidable uniformly in n and satisfies $x =_{n+1} x' \implies x =_n x'$. Moreover, $x = x'$ iff $s(x) = s(x')$, because s is injective, iff $s(x)(\delta_n) = s(x')(\delta_n)$ for every n, by density, iff $x =_n x'$ for every n, by definition. □

Proof of Theorem 6.1. The set $K_n = \{x \in X \mid f(x) =_n y\}$, being a decidable subset of an exhaustible space, is exhaustible. Therefore the result follows from Lemmas 6.3 and 6.4, because $x \in \bigcap_n K_n$ iff $f(x) =_n y$ for every n iff $f(x) = y$ by Lemma 6.5. □

Algorithms 6.6. In summary, the algorithm for semi-deciding non-existence of solutions is

$$\exists n.\forall x \in X.f(x) \neq_n y,$$

and that for computing the solution x_0 as a function of $\forall_X$, f, and y is:

$$\begin{aligned} \forall x \in K_n.p(x) \quad &= \quad \forall x \in X.f(x) =_n y \implies p(x), \\ \forall v \in L_n.q(v) \quad &= \quad \forall x \in K_n.q(s(x)), \\ n(z) \quad &= \quad \mu n.\forall v, w \in L_n.v(z) = w(z), \\ u(z) \quad &= \quad \mu m.\forall v \in L_{n(z)}.v(z) = m, \\ x_0 \quad &= \quad r(u). \end{aligned}$$

Here the parameters r and s come from the assumption that X is a kk-space. □

Remark 6.7. Even in the absence of uniqueness, *approximate* solutions with precision n are trivially computable with the algorithm

$$\varepsilon_X(\lambda x.f(x) =_n y),$$

using the fact that non-empty exhaustible subsets of kk-spaces are searchable. But the above unique-solution algorithm uses the quantification functional $\forall_X$ rather than the selection functional ε_X. In the next section, we compute solutions as limits of approximate solutions (cf. Remark 7.7). $\square$

7 Equations over metric spaces

We first formulate the main result of this section and then define the missing concepts. (Recall the notion of kk-exhaustibility defined in Section 5.3.)

Theorem 7.1. *Let X and Y be computational metric spaces with X computationally complete and kk-exhaustible.*

> *If $f\colon X \to Y$ and $y \in Y$ are computable, then, uniformly in f, y and the exhaustibility condition:*
>
> 1. *It is semi-decidable whether the equation $f(x) = y$ fails to have a solution $x \in X$.*
> 2. *If $f(x) = y$ has a unique solution $x \in X$, then it is computable.*

Hence any computable bijection $f\colon X \to Y$ has a computable inverse, uniformly in f and the exhaustibility condition.

There is a technical difficulty in the proof of the theorem: at the intensional level, where computations take place, solutions are unique only up to equivalence of representatives. In order to overcome this, we work with pseudo-metric spaces at the intensional level and with a notion of decidable closeness for them. Recall that a *pseudo-metric* on a set X is a function $d\colon X \times X \to [0, \infty)$ such that

1. $d(x, x) = 0$,
2. $d(x, y) = d(y, x)$,
3. $d(x, z) \leq d(x, y) + d(y, z)$.

Then d is a *metric* if it additionally satisfies $d(x, y) = 0 \implies x = y$. If d is only a pseudo-metric, then $(\sim)$ defined by

$$x \sim y \iff d(x, y) = 0$$

is an equivalence relation, referred to as *pseudo-metric equivalence*. A pseudo-metric topology is Hausdorff iff it is T_0 iff the pseudo-metric is a metric. Moreover, two points are equivalent iff they have the same neighbourhoods. Hence any sequence has at most one limit up to equivalence.

Definition 7.2. We regard $[0, \infty)$ as a computational space as in Example 4.6.

1. A *computational pseudo-metric space* is a computational space X endowed with a computable pseudo-metric, denoted by $d_X\colon X \times X \to [0, \infty)$ or simply d.

   We emphasize that we don't require the computational topology of X to agree with the pseudo-metric topology generated by open balls (cf. Remark 7.8).

2. A *computational metric space* is a computational pseudo-metric space in which the pseudo-metric is actually a metric, and hence we formulate the following definitions in the generality of pseudo-metric spaces.

3. A *fast Cauchy sequence* in a computational pseudo-metric space X is a sequence $x_n \in X$ with $d(x_n, x_{n+1}) < 2^{-n}$. The subspace of $X^{\mathbb{N}}$ consisting of fast Cauchy sequences is denoted by $\mathrm{Cauchy}(X)$.

4. A computational pseudo-metric space X is called *computationally complete* if every sequence $x_n \in \mathrm{Cauchy}(X)$ has a limit uniformly in x_n, i.e. there is a computable map $\lim\colon \mathrm{Cauchy}(X) \to X$ such that $\lim_n(x_n)$ is a limit of x_n for every sequence $x_n \in \mathrm{Cauchy}(X)$.

5. A computational pseudo-metric space X has *decidable closeness* if there is a family of relations $\sim_n$ on X that are decidable uniformly in n and satisfy:

   (a) $x \sim_n y \implies d(x,y) < 2^{-n}$,
   (b) $x \sim y \implies \forall_n . x \sim_n y$.
   (c) $x \sim_{n+1} y \implies x \sim_n y$,
   (d) $x \sim_n y \iff y \sim_n x$,
   (e) $x \sim_{n+1} y \sim_{n+1} z \implies x \sim_n z$.

   The last condition is a counter-part of the triangle inequality. It follows from the first condition that if $x \sim_n y$ for every n, then $x \sim y$. Write

   $$[x] = \{y \in X \mid x \sim y\}, \qquad [x]_n = \{y \in X \mid x \sim_n y\}.$$

   Then the equivalence class $[x]$ is the closed ball of radius zero centered at x.

We omit the qualification "computational" when it is clear that the context requires it. $\square$

For instance, the computational spaces $\mathbb{R}$ and $[0,\infty)$ defined in Example 4.6, are complete metric spaces in this computational sense under the Euclidean metric, as is well known, but don't have decidable closeness. We now proceed to prove the theorem.

Lemma 7.3. *For every computational metric space X there is a canonical computable pseudo-metric $d = d_{\ulcorner X \urcorner}$ on the representing space $\ulcorner X \urcorner$ such that:*

1. *The representation map $\rho = \rho_X : \ulcorner X \urcorner \to X$ is an isometry:*

   $$d(t,u) = d(\rho(t), \rho(u)).$$

   In particular:

   (a) $t \sim u \iff d(t,u) = 0 \iff \rho(t) = \rho(u)$.
   (b) If $f\colon X \to Y$ is a computable map of metric spaces, then any representative $\ulcorner f \urcorner\colon \ulcorner X \urcorner \to \ulcorner Y \urcorner$ preserves the relation $(\sim)$.

2. *If X is computationally complete then so is $\ulcorner X \urcorner$.*

3. *The representing space $\ulcorner X \urcorner$ has decidable closeness.*

Proof. Construct $d_{\ulcorner X \urcorner}\colon \ulcorner X \urcorner \times \ulcorner X \urcorner \to [0,\infty)$ as the composition of a computable representative $\ulcorner d_X \urcorner\colon \ulcorner X \urcorner \times \ulcorner X \urcorner \to \ulcorner [0,\infty) \urcorner$ of $d_X : X \times X \to [0,\infty)$ with the representation map $\rho_{[0,\infty)} : \ulcorner [0,\infty) \urcorner \to [0,\infty)$. A limit operator for $\ulcorner X \urcorner$ from a limit operator for X is constructed in a similar manner. For given $t, u \in \ulcorner X \urcorner$, let q_n be the n-th term of the sequence $\ulcorner d_X \urcorner(t,u) \in \ulcorner [0,\infty) \urcorner \subseteq \mathrm{Cauchy}(\mathbb{Q})$, and define $t \sim_n u$ to mean that the interval $[-2^{-n}, 2^{-n}]$ is contained in $[q_n - 2^{-n+1}, q_n + 2^{-n+1}]$. $\square$

We now work at the intensional level and later transfer the results to the extensional level with the aid of the above lemma.

Lemma 7.4. *Let Z be a complete computational pseudo-metric kk-space with decidable closeness, and $K_n \subseteq Z$ be a sequence of sets that are exhaustible uniformly in n and satisfy $K_n \supseteq K_{n+1}$.*

If $\bigcap_n K_n$ is an equivalence class, then it has a computable member,

uniformly in the given computational data.

Proof. Let $z \in \bigcap_n K_n$. For any m, we have $\bigcap_n K_n = [z] \subseteq [z]_{m+1}$, and hence there is n such that $K_n \subseteq [z]_{m+1}$, because the sets K_n are compact, because $K_n \supseteq K_{n+1}$, because Z is Hausdorff and because $[z]_{m+1}$ is open. Hence for every $u \in K_n$ we have $u \sim_{m+1} z$, and so for all $u, v \in K_n$ we have $u \sim_m v$. By exhaustibility of K_n and decidability of $(\sim_n)$, the function $n(m) = \mu n. \forall u, v \in K_n.\, u \sim_m v$ is computable. By searchability of K_n, there is a computable sequence $u_m \in K_{n(m)}$. Because $n(m) \leq n(m+1)$, we have that $K_{n(m)} \supseteq K_{n(m+1)}$ and hence $u_m \sim_m u_{m+1}$ and so $d(u_m, u_{m+1}) < 2^{-m}$ and u_m is a Cauchy sequence. By completeness, u_m converges to a computable point u_∞. Because $z \in K_{n(m)}$, we have $u_m \sim_m z$ for every m, and hence $d(u_m, z) < 2^{-m}$. And because $d(u_\infty, u_m) < 2^{-m+1}$, the triangle inequality gives $d(u_\infty, z) < 2^{-m} + 2^{-m+1}$ for every m and hence $d(u_\infty, z) = 0$ and therefore $u_\infty \in \bigcap_n K_n$. □

The proof of the following is essentially the same as that of Theorem 6.1, but uses Lemma 7.4 rather than Lemma 6.4, and doesn't rely on an analogue of Lemma 6.5, which is built into the definition of decidable closeness.

Lemma 7.5. *Let Z and W be pseudo-metric spaces with decidable closeness, and assume that Z is complete, exhaustible and a kk-space.*

If $g\colon Z \to W$ is a computable map that preserves pseudo-metric equivalence and $w \in W$ is computable, then, uniformly in $\forall_Z$, g, and w:

1. *It is semi-decidable whether the equivalence $g(z) \sim w$ fails to have a solution $z \in Z$.*
2. *If $g(z) \sim w$ has a unique solution $z \in Z$ up to equivalence, then some solution is computable.*

Proof. The set $K_n = \{z \in Z \mid g(z) \sim_n w\}$, being a decidable subset of an exhaustible space, is exhaustible. Therefore the result follows from Lemmas 6.3 and 7.4, because $z \in \bigcap_n K_n$ iff $g(z) \sim_n w$ for every n iff $g(z) = w$. □

Algorithm 7.6. The algorithm for computing the solution $z = u_\infty$ from $\forall_Z$, g and w is then the following, where we have expanded $\forall_{K_n}$ as a quantification over Z:

$$\begin{aligned} n(m) &= \mu n. \forall u, v \in Z.\, g(u) \sim_n w \wedge g(v) \sim_n w \implies u \sim_m v, \\ u_\infty &= \lim_m \varepsilon_K(\lambda z. g(z) \sim_{n(m)} w). \end{aligned}$$

Thus, although there are common ingredients with Theorem 6.1, the resulting algorithm is different, because it relies on the limit operator (but see Proposition 7.9 below). □

Remark 7.7. Again, approximate solutions are computable as in Remark 6.7, and, in fact, we are computing the unique solution as the limit of approximate solutions. But, for Theorem 7.1, approximate solutions are computable uniformly in $\ulcorner f \urcorner$ and $\ulcorner y \urcorner$ only, as different approximate solutions are obtained for different representatives of f and y. □

Proof of Theorem 7.1. Let X and Y be computational metric spaces with X kk-exhaustible, and let $f\colon X \to Y$ and $y \in Y$ be computable. Now apply Lemma 7.5 with $Z = \ulcorner X \urcorner$, $W = \ulcorner Y \urcorner$, $g = \ulcorner f \urcorner$, $w = \ulcorner y \urcorner$, using Lemma 7.3 to fulfill the necessary hypotheses. If $f(x) = y$ has a unique solution x, then $g(z) \sim w$ has a unique solution z up to equivalence, and $x = \rho(z)$ for any solution z, and hence x is computable. Because g preserves $(\sim)$ by Lemma 7.3, if $g(z) \sim w$ has a solution z, then $x = \rho(z)$ is a solution of $f(x) = y$. This shows that $f(x) = y$ has a solution iff $g(z) = w$ has a solution, and we can reduce the semi-decision of absence of solutions of $f(x) = y$ to absence of solutions of $g(z) = w$. $\square$

Before giving examples in computational real analysis, in Section 8, we clarify some aspects of the above development.

Remark 7.8. The metric topology of a computational space is always coarser than the computational topology, because, by continuity of the metric, open balls are open in the computational topology. Hence the computational topology of any computational metric space is Hausdorff. If X is kk-exhaustible and the metric topology is compact, then both topologies agree, because no compact Hausdorff topology can be properly refined to another compact Hausdorff topology. For e.g. the real line, the metric topology agrees with the computational topology, as is well-known, but in the example given in Proposition 7.9 below, it is strictly coarser. It is also strictly coarser for any pseudo-metric kk-space, because kk-spaces are Hausdorff and pseudo-metric topologies are Hausdorff only for pseudo-metrics which are genuine metrics. $\square$

Recall that an ultra-metric space is a metric space for which the triangle inequality holds in the stronger form $d(x, z) \leq \max(d(x, y), d(y, z))$, and that an ultra-metric topology is zero-dimensional because open balls are closed. Recall the equivalence relations $(=_n)$ given in Lemma 6.5.

Proposition 7.9. *Any kk-space X equipped with d defined by*

$$d(x, y) = \inf\{2^{-n} \mid x =_n y\}$$

is a computational ultra-metric space. Moreover:

1. *The metric has decidable closeness given by $(\sim_n) = (=_n)$.*
2. *The metric topology is in general strictly coarser than the computational topology, but both agree on compact subsets.*
3. *Exhaustible subspaces with the relative metric are computationally complete.*

Proof. Computability of the ultra-metric and decidability of closeness are easy.

Let $s\colon X \to \mathbb{N}^Z$ and $r\colon \mathbb{N}^Z \to X$ be the same computable functions selected in the proof of Lemma 6.5, let δ be the same dense sequence, and for $u, v \in \mathbb{N}^Z$ define $u =_n v \iff \forall i < n.\, u(\delta_i) = v(\delta_i)$ so that $x =_n y$ iff $s(x) =_n s(y)$.

Agreement of topologies: A subbasic open set in the topology of pointwise convergence of $\mathbb{N}^Z$ is of the form $N(z, V) = \{u \in \mathbb{N}^Z \mid u(z) \in V\}$ with $z \in Z$ and $V \subseteq \mathbb{N}$. Now $u =_n v$ iff $d(u, v) < 2^{-n}$, and hence the open ball $B_{2^{-n}}(u)$ is the intersection of the pointwise open sets $N(\delta_i, \{u(\delta_i)\})$, for $i < n$, and hence open balls are open in the topology of pointwise convergence. For a compact subspace of $\mathbb{N}^Z$, density of δ gives that metric topology agrees with the pointwise topology. But the relative topology on compact subsets of $\mathbb{N}^Z$ coincides with the topology of pointwise convergence, by classical, and easy, Arzela–Ascoli type arguments, and hence the three topologies agree. The reduction of this to X via the retraction is easy.

Completeness: Let $z_n \in Z$ be a fast Cauchy sequence. Then $z_n =_n z_{n+1}$ and hence $s(z_n) =_n s(z_{n+1})$. It suffices to show that the sequence $f_n = s(z_n)$ converges to a computable limit f_∞, because then the sequence $z_n = r(f_n)$ converges to the computable point $z_\infty = r(f_\infty)$ by continuity of r. The set $L = s(K)$ is exhaustible because it is a computable image of an exhaustible set. For any n, the set $L_n = \{g \in L \mid g =_n f_n\}$ is exhaustible because it is a decidable subset of an exhaustible set, and $f_n \in L_n$. By compactness, $\bigcap_n L_n \neq \emptyset$ because clearly $L_n \supseteq L_{n+1}$. If $g, h \in \bigcap_n L_n$, then $g =_n f_n =_n h$ for every n, and hence $g = h$, and so $\bigcap_n L_n = \{f_\infty\}$ for some computable f_∞ by Lemma 6.4. Because $f_\infty \in \bigcap_n L_n$, we have $f_\infty =_n f_n$ for every n. Hence if some ball $B_{2^{-k}}(h)$ is a neighbourhood of f_∞, then $h =_k f_\infty =_k f_n$ for all $n \geq k$, and hence $f_n \in B_k(h)$ for all $n \geq k$, which shows that $f_n \to f_\infty$. □

In view of this proposition, Lemma 7.4 generalizes Lemma 6.4. But Lemma 6.4 cannot be eliminated, because it is used to prove the proposition.

Algorithm 7.10. Expanding Lemma 6.4, the algorithm for computing $\lim_n f_n$ for a fast Cauchy sequence $f_n \in L \subseteq \mathbb{N}^Z$ with L exhaustible is:

$$\begin{aligned} n(z) &= \mu n. \forall g, h \in L. g =_n f_n \wedge h =_n f_n \implies g(z) = h(z), \\ \lim_n f_n &= \lambda z. \mu m. \forall g \in L. g =_{n(z)} f_{n(z)} \implies g(z) = m. \end{aligned}$$

Notice that "fast" amounts to $f_n =_n f_{n+1}$. □

Remark 7.11. Independently of this, Matthias Schröder (personal communication) showed that if a QCB space X is the sequential coreflection of a zero-dimensional topology, then there is a metric d on X such that: (1) The topology induced by d is coarser than that of X and than the zero-dimensional topology. (2) On compact subsets X, the three topologies agree. (3) The image of d is $\{0\} \cup \{2^{-n} | n \in \mathbb{N}\}$. This applies to all kk-spaces in particular, as their topologies satisfy the hypothesis. His construction uses countable pseudo-bases rather than dense sequences. However, he hasn't considered the computational versions of these statements. □

8 Exhaustible spaces of analytic functions

For any $\epsilon \in (0,1)$, any $x \in [-\epsilon, \epsilon]$, any $b > 0$, and any sequence $a \in [-b,b]^{\mathbb{N}}$, the Taylor series $\sum_n a_n x^n$ converges to a number in the interval $[-b/(1+\epsilon), b/(1-\epsilon)]$.

Lemma 8.1. *Any analytic function $f \in \mathbb{R}^{[-\epsilon,\epsilon]}$ of the form*

$$f(x) = \sum_n a_n x^n$$

is computable uniformly in any given computable $\epsilon \in (0,1)$, $b > 0$ and $a \in [-b,b]^{\mathbb{N}}$.

Proof. Standard computational analysis argument. □

Definition 8.2. Denote by

$$A = A(\epsilon, b)$$

the subspace of such analytic functions and by

$$T = T_{\epsilon,b} \colon [-b,b]^{\mathbb{N}} \to A(\epsilon, b)$$

the functional that implements the uniformity condition, so that $f = T(a)$. □

The following results also hold uniformly in ϵ and b, but we omit explicit indications. Also, the results are uniform in the exhaustibility assumptions.

Theorem 8.3. *The space A is kk-exhaustible.*

Proof. The space $[-b, b]^{\mathbb{N}}$ is kk-exhaustible by Examples 5.7. By Lemma 4.8, any computable representative $\ulcorner T \urcorner \colon \ulcorner [-b, b]^{\mathbb{N}} \urcorner \to \ulcorner A \urcorner$ is a surjection, and hence $\ulcorner A \urcorner$ is exhaustible. It follows from Lemma 5.5 that $\ulcorner A \urcorner$ is a kk-space, because the representing space $\ulcorner A \urcorner$ is clearly a subspace of a Kleene–Kreisel space. $\square$

Hence the solution of a functional equation with a unique analytic unknown in A can be computed using Theorem 7.1.

Lemma 8.4. *For any non-empty kk-exhaustible space X, the maximum- and minimum-value functionals*

$$\max_X, \min_X \colon \mathbb{R}^X \to \mathbb{R}$$

are computable.

NB. Of course, any $f \in \mathbb{R}^X$ attains its maximum value because it is continuous and because kk-exhaustible spaces are compact.

Proof. We discuss $\max$ only. By e.g. the algorithm given by Simpson [6], this is the case for $X = 2^{\mathbb{N}}$. Because the representing space $\ulcorner X \urcorner$, being a non-empty exhaustible kk-space, is a computable image of the Cantor space, the space X itself is a computable image of the cantor space, say with $q \colon 2^{\mathbb{N}} \to X$. Then the algorithm $\max_X(f) = \max_{2^N}(f \circ q)$ gives the required conclusion. $\square$

Corollary 8.5. *Any kk-exhaustible subspace K of a metric space X is computably located in the sense that the distance function $d_K \colon X \to \mathbb{R}$ defined by*

$$d_K(x) = \min\{d(x, y) \mid y \in K\}$$

is computable.

Proof. $d_K(x) = \min_K(\lambda y.d(x, y))$. $\square$

Corollary 8.6. *For any non-empty kk-exhaustible metric space X, the max-metric on $\mathbb{R}^X$*

$$d(f, g) = \max\{d(f(x), g(x)) \mid x \in X\}$$

is computable.

Proof. $d(f, g) = \max_X(\lambda x.d(f(x), g(x)))$. $\square$

Corollary 8.7. *For $f \in \mathbb{R}^X$, it is semi-decidable whether $f \notin A$.*

Proof. Because A is computationally located in $\mathbb{R}^{[-\epsilon,\epsilon]}$ as it is kk-exhaustible, and because $f \notin A \iff d_A(f) \neq 0$. $\square$

Another proof, which doesn't rely on the kk-exhaustibility of A, uses Theorem 7.1: $f \notin A$ iff the equation $T(a) = f$ doesn't have a solution $a \in [-b, b]^{\mathbb{N}}$. But this alternative proof relies on a complete metric on $[-b, b]^{\mathbb{N}}$. For simplicity, we consider a standard construction for 1-bounded metric spaces. This is no loss of generality for our purposes, because for any metric d, the metric $d'(x, y) = \min(1, d(x, y))$ has the same Cauchy sequences. (Moreover, because we shall confine our attention to kk-exhaustible metric spaces, this is no loss of generality with the alternative reason that the diameter of such a space is computable as $\max(\lambda x. \max(\lambda y.d(x, y)))$.)

Lemma 8.8. *For any computational* 1*-bounded metric space* X*, the metric on* $X^{\mathbb{N}}$ *defined by*

$$d(x, y) = \sum_n 2^{-n-1} d(x_n, y_n)$$

is computable and 1*-bounded, and it is computationally complete if* X *is.*

Proof. Use the fact that the map $[0, 1]^{\mathbb{N}} \to [0, 1]$ that sends a sequence $a \in [0, 1]^{\mathbb{N}}$ to the number $\sum_n 2^{-n-1} a_n$ is computable. Regarding completeness, it is well known that a sequence in the space $X^{\mathbb{N}}$ is Cauchy iff it is componentwise Cauchy in X, and in this case its limit is calculated componentwise. (To compute the limit componentwise, maybe we need suitable shifting to make sure that the components converge at the required speed — I have to check this.) □

Corollary 8.9. *The Taylor coefficients of any* $f \in A$ *can be computed from* f*.*

Proof. The space $[-b, b]^{\mathbb{N}}$ is kk-exhaustible by Examples 5.7, and hence the function T is invertible by Theorem 7.1 and Lemma 8.8. □

Now, it remains to write down: analytic functions of the form $\sum_n a_n x^n / n!$, with the same restrictions on a and ϵ, are also kk-exhaustible, and are closed under differentiation, and they are computably differentiable. Then it remains to think whether this can be used to solve differential equations using e.g. the classical Peano theorem. In certain cases, it is known that analytic solutions are unique, but this uses the complex plane. It should be easy to generalize the above results to complex analytic functions. Can the results discussed above be developed in the internal logic of the category and then be extracted via realizability? Work on this paper will have to be suspended for at least a month, unfortunately.

References

[1] A. Bauer. *The Realizability Approach to Computable Analysis and Topology*. PhD thesis, Carnegie Mellon University, School of Computer Science, 2000. CMU technical report CMU-CS-00-164.

[2] A. Bauer. A relationship between equilogical spaces and type two effectivity. *MLQ Math. Log. Q.*, 48(suppl. 1):1–15, 2002. Dagstuhl Seminar on Computability and Complexity in Analysis, 2001.

[3] A. Bauer, L. Birkedal, and D.S. Scott. Equilogical spaces. *Theoret. Comput. Sci.*

[4] M.J. Beeson. *Foundations of Constructive Mathematics*. Springer, 1985.

[5] M. Escardó. Exhautible sets in higher-type computation. *Logical Methods in Computer Science*, 2008. to appear.

[6] A. Simpson. Lazy functional algorithms for exact real functionals. *Lec. Not. Comput. Sci.*, 1450:323–342, 1998.