De Groot duality for represented spaces

Takayuki Kihara and Arno Pauly

[1] Department of Mathematical Informatics
Nagoya University, Japan
kihara@i.nagoya-u.ac.jp
[2] School of Mathematics & Computer Science
Swansea University, Swansea, United Kingdom
Arno.Pauly@gmail.com

Abstract. We explore de Groot duality in the setting of represented spaces. The de Groot dual of a space is the space of closures of its singletons, with the representation inherited from the hyperspace of closed subsets. This yields an elegant duality, in particular between Hausdorff spaces and compact T_1-spaces. As an application of the concept, we study the point degree spectrum of the dual of Baire space, and show that it is, in a formal sense, far from being countably-based.

1 Introduction

In this article, through the theory of represented spaces and higher type computability, we give a unified treatment of the studies of Π^0_1 *singletons* in classical computability theory [8, Definition XII.2.13] and *de Groot duality* in general topology [3, Section 9.1.2]. The former notion has been associated with *implicit definability* in classical logic [8, Definition XII.2.13]; hence, this unified treatment gives de Groot duality a new interpretation: the duality of "explicit" and "implicit". Conversely, the pure topological aspect of the latter also provides a renewed understanding of Π^0_1 singletons. By exploring these notions, in this article, we see an elegant duality between Hausdorff spaces and compact T_1-spaces.

Formally, we introduce the de Groot dual of a represented space. Recall that for any represented space $\mathbf{X}$, we obtain the represented space $\mathcal{A}(\mathbf{X})$ of closed subsets by identifying a set with the characteristic function of its complement into Sierpiński space.

Definition 1. *For a represented space* $\mathbf{X}$*, let* $\mathbf{X}^{\mathsf{d}}$ *denote the space* $\{\overline{\{x\}} \mid x \in \mathbf{X}\} \subseteq \mathcal{A}(\mathbf{X})$*. We call* $\mathbf{X}^{\mathsf{d}}$ *the* de Groot dual *of* $\mathbf{X}$*.*

Example 2. Computable points in $(\mathbb{N}^{\mathbb{N}})^{\mathsf{d}}$ are exactly Π^0_1 singletons in $\mathbb{N}^{\mathbb{N}}$.

Usually, we are only interested in T_0 represented spaces, and we will assume spaces to be T_0 throughout the rest of the paper[3]. The T_0-property is equivalent

[3] The de Groot dual of a space is the same as the de Groot dual of its T_0-quotient anyway.

to $x \mapsto \overline{\{x\}} : \mathbf{X} \to \mathbf{X}^{\mathsf{d}}$ being a bijection, and we can thus treat $\mathbf{X}$ and $\mathbf{X}^{\mathsf{d}}$ to have the same underlying set. The de Groot dual is particularly well-behaved when we restrict our attention further to T_1-spaces, where points are already closed. A primary appeal of the dual is that for T_1-spaces, it interchanges Hausdorff and compact spaces. We summarize the properties of de Groot duality for T_1-spaces in Theorem 3 in Section 2.

While the de Groot dual has a natural definition in the setting of represented spaces, the concept is originally from topology [4]; see [3, Section 9.1.2]. For a topological space $\mathcal{X}$, its *de Groot dual* is the topology on $\mathcal{X}$ generated by complements of saturated compact subsets of $\mathcal{X}$. It is no surprise to have an analogy between concepts for represented spaces and topological spaces [2, 9]; and each represented space naturally comes equipped with a topology. Often, these concepts align (only) up to sequentialization. We leave the study of the precise relation of de Groot duality for represented spaces and for topological spaces for future work.

1.1 Preliminaries

We briefly recap some preliminaries, and refer to [9] for more details. A *represented space* is a set X equipped with a partial surjection $\delta_X \colon\subseteq \mathbb{N}^{\mathbb{N}} \to X$. If $\delta_X(p) = x$ then we say that p is a name of x. A point $x \in \mathbf{X}$ is computable if it has a computable name. A function $f \colon \mathbf{X} \to \mathbf{Y}$ is continuous (computable, resp.) if there exists a continuous (computable, resp.) function which, given a name of $x \in \mathbf{X}$, returns a name of $f(x) \in \mathbf{Y}$. We write $\mathbf{X} \simeq \mathbf{Y}$ if $\mathbf{X}$ is computably isomorphic to $\mathbf{Y}$. One of the remarkable properties of the category of represented spaces and continuous (computable) functions is that it is cartesian closed.

We denote the represented *Sierpiński space* by $\mathbb{S}$, which consists of a closed point $\bot$ (whose name is $000\dots$) and an open point $\top$ (whose names are other sequences). A subset A of a represented space $\mathbf{X}$ is *open* if its characteristic map $\chi_A \colon \mathbf{X} \to \mathbb{S}$ is continuous. Identifying a subset with its characteristic map, the represented hyperspace $\mathcal{O}(\mathbf{X})$ of all open subsets of $\mathbf{X}$ can be defined by the exponential $\mathbb{S}^{\mathbf{X}}$. In a similar way, the represented hyperspace $\mathcal{A}(\mathbf{X})$ of all closed sets can also be defined. As a subspace of a represented space is also represented, the de Groot dual $\mathbf{X}^{\mathsf{d}} \subseteq \mathcal{A}(\mathbf{X})$ can also be treated as a represented space. If there is no risk of confusion, a point $\overline{\{x\}}$ in the de Groot dual $\mathbf{X}^{\mathsf{d}}$ is simply written as x. Formally, this is justified by defining a dual name of x as an $\mathbf{X}^{\mathsf{d}}$-name of $\overline{\{x\}}$.

Given $x \in \mathbf{X}$, let $\kappa_{\mathbf{X}}(x)$ be the neighborhood filter of x; that is, $\{U \in \mathcal{O}(\mathbf{X}) : x \in U\}$. Note that $\kappa_{\mathbf{X}} \colon \mathbf{X} \to \mathcal{O}\mathcal{O}(\mathbf{X})$ is always well-defined and computable, since $\kappa_{\mathbf{X}}$ is obtained as currying of the evaluation map $\in \colon X \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$. A space $\mathbf{X}$ is *computably admissible* if $\kappa_{\mathbf{X}}$ has a partial computable left-inverse. The image $\mathbf{X}_\kappa$ of $\kappa_{\mathbf{X}}$ is called the *admissibilification* of $\mathbf{X}$. Note that $\mathbf{X}$ is computably admissible iff $\kappa_{\mathbf{X}}$ is a computable isomorphism between $\mathbf{X}$ and $\mathbf{X}_\kappa$.

A space $\mathbf{X}$ is *computably compact* if $\forall_{\mathbf{X}} \colon \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ is computable, where $\forall_{\mathbf{X}}(U) = \top$ iff $U = X$. Equivalently, $\{U \in \mathcal{O}(\mathbf{X}) : \mathbf{U} = \mathbf{X}\}$ is a computable point in $\mathcal{O}\mathcal{O}(\mathbf{X})$. A space $\mathbf{X}$ is *computably Hausdorff* if $\neq \colon \mathbf{X} \times \mathbf{X} \to \mathbb{S}$ is computable.

A space $\mathbf{X}$ is T_1 if, for any $x \in \mathbf{X}$, $\neq_x : \mathbf{X} \to \mathbb{S}$ defined by $\neq_x(y) = (x \neq y)$ is continuous; that is, $\{x\} \in \mathcal{A}(\mathbf{X})$. A space $\mathbf{X}$ is T_0 if $\kappa_\mathbf{X}$ has a partial left-inverse. Note that these notions are defined for a represented space (not necessarily a topological space), although, of course, a represented space can always be equipped with its quotient topology.

2 Duality for T_1 represented spaces

If we restrict to T_1 spaces, the de Groot dual of $\mathbf{X}$ is simply the space of closed singletons of $\mathbf{X}$, with the subspace representation inherited from $\mathcal{A}(\mathbf{X})$. It is in this setting that the dual exhibits very elegant properties, and in particular becomes a duality between Hausdorffness and compactness. The following theorem lays out how the duality works. The proofs of its claims are spread throughout Subsection 2 below. The requirement for the space and or its dual to contain one or two computable points are used only for a few of the implications. We do not know whether these requirements are needed, but having some computable points seems like a sufficiently innocent restriction.

Theorem 3. *Let $\mathbf{X}$ be computably admissible and T_1, and let $\mathbf{X}$ and $\mathbf{X}^\mathsf{d}$ each contain two computable points. Then:*

1. $\mathrm{id} : \mathbf{X} \to \mathbf{X}^\mathsf{dd}$ *is computable.*
2. $\mathbf{X}^\mathsf{d} \cong \mathbf{X}^\mathsf{ddd}$.
3. *The following are equivalent:*
   (a) $\mathbf{X}$ *is computably Hausdorff.*
   (b) $\mathbf{X}$ *is computably Hausdorff and* $\mathbf{X} \cong \mathbf{X}^\mathsf{dd}$.
   (c) $\mathbf{X}^\mathsf{dd}$ *is computably Hausdorff.*
   (d) $\mathbf{X}^\mathsf{d}$ *is computably compact.*
   (e) $\mathrm{id} : \mathbf{X} \to \mathbf{X}^\mathsf{d}$ *is computable.*
   (f) $\mathrm{id} : \mathbf{X}^\mathsf{dd} \to \mathbf{X}^\mathsf{d}$ *is computable.*
4. *The following are equivalent:*
   (a) $\mathbf{X}$ *is computably compact.*
   (b) $\mathbf{X}^\mathsf{dd}$ *is computably compact.*
   (c) $\mathbf{X}^\mathsf{d}$ *is computably Hausdorff.*
   (d) $\mathrm{id} : \mathbf{X}^\mathsf{d} \to \mathbf{X}$ *is computable.*
   (e) $\mathrm{id} : \mathbf{X}^\mathsf{d} \to \mathbf{X}^\mathsf{dd}$ *is computable.*
5. *The following are equivalent:*
   (a) $\mathbf{X}$ *is computably compact and computably Hausdorff.*
   (b) $\mathbf{X} \cong \mathbf{X}^\mathsf{d}$.

The item (5) can be thought of as a computable version of [7, Example 4.1]. While the situation of Hausdorffness and compactness are mostly symmetrical in our main theorem, there is a notable absence: For computably compact $\mathbf{X}$ we cannot conclude that $\mathbf{X} \cong \mathbf{X}^\mathsf{dd}$. An example for this is exhibited in Section 4.

Proofs of the basics. We proceed to prove the various components of Theorem 3. Most of the proofs are presented by crystal-clear arguments based on higher type computability. These seem to fit well with synthetic topology [2], with the exception of the proofs of Propositions 8 and 16 and Lemma 14.

Throughout this section, the space $\mathbf{X}$ is assumed to be T_1 without this being necessarily stated explicitly.

Observation 4 *The map* $\neq : \mathbf{X} \times \mathbf{X}^{\mathsf{d}} \to \mathbb{S}$ *is computable.*

Proof. As $\mathcal{A}(\mathbf{X}) \simeq \mathbb{S}^{\mathbf{X}}$, the non-membership relation $\notin : \mathbf{X} \times \mathcal{A}(\mathbf{X}) \to \mathbb{S}$ is exactly the evaluation map, so it is computable. For $x, y \in X$, note that $x \notin \{y\}$ iff $x \neq y$. Thus, the non-membership relation $\notin$ restricted to $\mathbf{X} \times \mathbf{X}^{\mathsf{d}}$ is exactly the non-equality relation $\neq$ via the identification of $\{y\}$ with $y \in \mathbf{X}^{\mathsf{d}}$. Therefore, $\neq : \mathbf{X} \times \mathbf{X}^{\mathsf{d}} \to \mathbb{S}$ is computable. □

Corollary 5. *1.* $\mathbf{X}^{\mathsf{d}}$ *is* T_1 *(and thus* $\mathbf{X}^{\mathsf{dd}}$ *is well-defined).*
2. $\mathrm{id} : \mathbf{X} \to \mathbf{X}^{\mathsf{dd}}$ *is computable.*

Proof. For (1), currying the function $\neq$ in Observation 4 yields the function $x \mapsto X \setminus \{x\} : \mathbf{X} \to \mathcal{O}(\mathbf{X}^{\mathsf{d}})$. In particular, $X \setminus \{x\}$ is open in $\mathbf{X}^{\mathsf{d}}$, which means that $\mathbf{X}^{\mathsf{d}}$ is T_1. For (2), as currying preserves computability, the above function is computable, and an $\mathcal{O}(\mathbf{X}^{\mathsf{d}})$-name of $X \setminus \{x\}$ is exactly an $\mathbf{X}^{\mathsf{dd}}$-name of x. □

The following is essentially just a rephrasing of the definition of being computably Hausdorff:

Observation 6 $\mathrm{id} : \mathbf{X} \to \mathbf{X}^{\mathsf{d}}$ *is computable iff* $\mathbf{X}$ *is computably Hausdorff.*

Proof. As in Corollary 5, one can see that $\mathrm{id} : \mathbf{X} \to \mathbf{X}^{\mathsf{d}}$ is computable iff $\neq : X \times X \to \mathbb{S}$ is computable, which means that $\mathbf{X}$ is computably Hausdorff. □

The connection to unique closed choice. The map $\mathrm{id} : \mathbf{X}^{\mathsf{d}} \to \mathbf{X}$ is just another perspective on the principle of *unique closed choice* $\mathrm{UC}_{\mathbf{X}}$ studied in [1], which is formally a partial function $\mathrm{UC}_{\mathbf{X}} :\subseteq \mathcal{A}(\mathbf{X}) \to \mathbf{X}$, whose domain is the set of all closed singletons, and $\mathrm{UC}_{\mathbf{X}}(\{a\}) = a$ for any $a \in X$. In particular, $\mathrm{id} : \mathbf{X}^{\mathsf{d}} \to \mathbf{X}$ is computable iff $\mathrm{UC}_{\mathbf{X}}$ is computable. More or less by the definition of admissibility, we find that $\mathrm{UC}_{\mathbf{X}}$ is computable for a computably compact computably admissible space:

Observation 7

1. *If* $\mathbf{X}$ *is computably compact and computably admissible, then* $\mathrm{id} : \mathbf{X}^{\mathsf{d}} \to \mathbf{X}$ *is computable.*
2. *If* $\mathbf{X}$ *is computably compact, then* $\mathbf{X}^{\mathsf{d}}$ *is computably Hausdorff.*

Proof. (1) A name of a given $x \in \mathbf{X}^{\mathsf{d}}$ is also a name of $X \setminus \{x\} \in \mathcal{O}(\mathbf{X})$. By computable compactness, given $U \in \mathcal{O}(\mathbf{X})$, one can semidecide if $(X \setminus \{x\}) \cup U =$

X, which is true iff $x \in U$. By computable admissibility, this yields an $\mathbf{X}$-name of x.

(2) Names of $x, y \in \mathbf{X}^{\mathsf{d}}$ are also names of $X \setminus \{x\}, X \setminus \{y\} \in \mathcal{O}(\mathbf{X})$. By computable compactness, one can semidecide if $(X \setminus \{x\}) \cup (X \setminus \{y\}) = X$, which is true iff $x \neq y$. This shows that $\neq\colon \mathbf{X}^{\mathsf{d}} \times \mathbf{X}^{\mathsf{d}} \to \mathbb{S}$ is computable. Consequently, $\mathbf{X}^{\mathsf{d}}$ is computably Hausdorff. ☐

Observation 7 (1) generalizes the classical observation that a Π^0_1 singleton in Cantor space is computable [8, Exercise XII.2.15 (c)] (whose uniform version is given in [1, Corollary 6.4] in the context of a unique closed choice).

Interestingly, we also have a converse direction, which gives a topological interpretation of the classical observation that a Π^0_1 singleton in Baire space (which is non-compact) is not necessarily computable [8, Exercise XII.2.15 (d)]. The statement can be described using the Weihrauch degree of $\mathrm{UC}_{\mathbf{X}}\colon \{a\} \mapsto a$. That $\mathbf{1} \leq_{\mathrm{W}} \mathrm{UC}_{\mathbf{X}}$ means that $\mathrm{UC}_{\mathbf{X}}$ has a computable instance $\{a\} \in \mathcal{A}(\mathbf{X})$. That $\mathrm{UC}_{\mathbf{X}} \leq_{\mathrm{W}} \mathbf{1}$ just means that $\mathrm{UC}_{\mathbf{X}}$ is computable.

Proposition 8. *If* $\mathrm{UC}_{\mathbf{X}} \equiv_{\mathrm{W}} \mathbf{1}$*, then* $\mathbf{X}$ *is computably compact.*

Before we begin the proof, let us make a technical comment. For $A, B \in \mathcal{O}(\mathbf{X})$, one can see that $A \subseteq B$ iff $A \leq_{\mathcal{O}(\mathbf{X})} B$; that is, A is contained in the closure of $\{B\}$ in $\mathcal{O}(X)$. In fact, the standard representation of the function space $\mathcal{O}(\mathbf{X}) \simeq \mathbb{S}^{\mathbf{X}}$ gives us even a better property: If $A \subseteq B$ then the set of names of A is included in the closure of the set of names of B; that is, any neighborhood of a name of A contains a name of B.

Proof (Proposition 8). To prove that $\mathbf{X}$ is computably compact, we need to prove that given some $U \in \mathcal{O}(\mathbf{X})$ we can recognize if $U = X$. To do this, we compute $U^a := \{a\} \cup (X \setminus U) \in \mathcal{A}(\mathbf{X})$, and attempt to semidecide $\mathrm{UC}_{\mathbf{X}}(U^a) \in U$?. If we get a positive answer, we conclude that $U = X$. Note that since U^a is not necessarily in the domain of $\mathrm{UC}_{\mathbf{X}}$, this is not a well-typed expression - we just run it as a partial algorithm to the best of our ability.

For correctness of this algorithm, first consider the case that $X = U$. Then $U^a = \{a\}$, and thus $\mathrm{UC}_{\mathbf{X}}(U^a)$ is well-defined and returns a, and $a \in U = X$ is going to be recognized as true. Next, we consider the case that $U = X \setminus \{a\}$. Again, we have that $U^a = \{a\}$, and $\mathrm{UC}_{\mathbf{X}}(U^a) = a$, but as $a \notin U$, we will not answer *yes*. Finally, we consider the case where there exists some $b \neq a$ with $b \notin U$. Since $\{b\} \subseteq U^a \in \mathcal{A}(\mathbf{X})$, the name for U^a we compute can change arbitrarily late to be a name for $\{b\}$ instead; that is, any finite prefix of a name of U^a can be extended to a name of $\{b\}$. While the computation $\mathrm{UC}_{\mathbf{X}}(U^a)$ has no need to output anything, it must not output a prefix which cannot be extended to a name for b. But if a prefix returned by $\mathrm{UC}_{\mathbf{X}}(U^a)$ is sufficient to confirm membership of the potential output in U, it is inconsistent with a name for b, as $b \notin U$. Thus, this case cannot lead to an erroneous positive answer. ☐

Corollary 9. *If* $\mathrm{id} : \mathbf{X}^{\mathsf{d}} \to \mathbf{X}$ *is computable and* $\mathbf{X}^{\mathsf{d}}$ *contains a computable point, then* $\mathbf{X}$ *is computably compact.*

Proof. Just note that id: $\mathbf{X}^{\mathsf{d}} \to \mathbf{X}$ is computable iff $\mathrm{UC}_{\mathbf{X}} \leq_{\mathrm{W}} \mathbf{1}$, and $\mathbf{X}^{\mathsf{d}}$ contains a computable point iff $\mathbf{1} \leq_{\mathrm{W}} \mathrm{UC}_{\mathbf{X}}$. Then the assertion follows from Proposition 8. □

Corollary 10. *Let* $\mathbf{X}$ *contain a computable point. Then the following are equivalent:*

1. $\mathrm{id} : \mathbf{X} \to \mathbf{X}^{\mathsf{d}}$ *and* $\mathrm{id} : \mathbf{X}^{\mathsf{d}} \to \mathbf{X}$ *are both computable.*
2. $\mathbf{X}$ *is computably admissible, computably compact and computably Hausdorff.*

Proof. The direction from (2) to (1) follows from Observations 6 and 7 (1). For the direction from (1) to (2), Observation 6 and Corollary 9 show that $\mathbf{X}$ is computably compact and computably Hausdorff. It remains to show that $\mathbf{X}$ is computably admissible. Let $\mathrm{ev}_x \colon \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ defined by $\mathrm{ev}_x(U) = (x \in U)$ be given. As $y \mapsto X \setminus \{y\} \colon \mathbf{X}^{\mathsf{d}} \to \mathcal{O}(\mathbf{X})$ is a computable embedding, the function $y \mapsto \mathrm{ev}_x(X \setminus \{y\}) \colon \mathbf{X}^{d} \to \mathbb{S}$ is also computable. Note that $\mathrm{ev}_x(X \setminus \{y\}) = \top$ iff $x \neq y$, so this yields a name of $X \setminus \{x\} \in \mathcal{O}(\mathbf{X}^{\mathsf{d}})$, which is exactly an $\mathbf{X}^{\mathsf{dd}}$-name of x. This shows that $\mathrm{ev}_x \mapsto x \colon \subseteq \mathcal{O}\mathcal{O}(\mathbf{X}) \to \mathbf{X}^{\mathsf{dd}}$ is always computable. By using our assumption (1) twice, we see that id: $\mathbf{X}^{\mathsf{dd}} \to \mathbf{X}$ is computable, so we conclude that $\mathbf{X}$ is computably admissible. □

More on Hausdorffness.

Proposition 11. *If* $\mathbf{X}$ *is computably Hausdorff, then* id: $\mathbf{X}^{\mathsf{dd}} \to \mathbf{X}^{\mathsf{d}}$ *is computable.*

Proof. By Observation 4, $\neq \colon \mathbf{X}^{\mathsf{d}} \times \mathbf{X}^{\mathsf{dd}} \to \mathbb{S}$ is computable. Since $\mathbf{X}$ is computably Hausdorff, by Observation 6, id: $\mathbf{X} \to \mathbf{X}^{\mathsf{d}}$ is computable, so $\neq \colon \mathbf{X} \times \mathbf{X}^{\mathsf{dd}} \to \mathbb{S}$ is computable. By currying, the function $x \mapsto X \setminus \{x\} \colon \mathbf{X}^{\mathsf{dd}} \to \mathcal{O}(\mathbf{X})$ is computable, which means that id: $\mathbf{X}^{\mathsf{dd}} \to \mathbf{X}^{\mathsf{d}}$ is computable. □

Corollary 12. *Let* $\mathbf{X}$ *be computably Hausdorff and contain a computable point. Then* $\mathbf{X}^{\mathsf{d}}$ *is computably compact.*

Proof. By Proposition 11 and Corollary 9 (applied to $\mathbf{X}^{\mathsf{d}}$ rather than $\mathbf{X}$). Note that by Corollary 5 (2), we obtain a computable point in $\mathbf{X}^{\mathsf{dd}}$ from the one we have in $\mathbf{X}$. □

Corollary 13. *If* $\mathbf{X}^{\mathsf{d}}$ *is computably compact, then* $\mathbf{X}$ *is computably Hausdorff.*

Proof. By Observation 7 (2), if $\mathbf{X}^{\mathsf{d}}$ is computably compact, then $\mathbf{X}^{\mathsf{dd}}$ is computably Hausdorff. By Corollary 5, id: $\mathbf{X} \to \mathbf{X}^{\mathsf{dd}}$ is computable, so $\mathbf{X}$ admits a computable injection into a computable Hausdorff space, and is thus itself computably Hausdorff. □

Lemma 14. *Let* $\mathbf{X}^{\mathsf{d}}$ *contain two computable points and let* $\mathbf{X}$ *be computably admissible. Then* $\mathrm{id} : (\mathbf{X}^{\mathsf{d}} \wedge \mathbf{X}^{\mathsf{dd}}) \to \mathbf{X}$ *is computable.*

Proof. We are given $\{x\} \in \mathbf{X}^{\mathsf{d}}$ and $\{\{x\}\} \in \mathbf{X}^{\mathsf{dd}}$ and seek to compute $x \in \mathbf{X}$. As $\mathbf{X}$ is computably admissible, we can equivalently seek to semidecide whether $x \in U$ for given $U \in \mathcal{O}(\mathbf{X})$. In addition, we have access to computable $\{y\}, \{z\} \in \mathbf{X}^{\mathsf{d}}$ with $y \neq z$.

We can compute $A_y = \{y\} \cup (\{x\} \cap (X \setminus U))$ and $A_z = \{z\} \cup (\{x\} \cap (X \setminus U))$ as elements of $\mathcal{A}(\mathbf{X})$. While A_y and A_z may fail to be singletons, and thus the queries $A_y \in \{\{x\}\}$? and $A_z \in \{\{x\}\}$? are not necessarily well-typed, we can attempt the computations anyway. If either of them yields a *no*-answer, we have confirmed that $x \in U$.

To see this, first consider the case where $x \in U$. Then $A_y = \{y\}$ and $A_z = \{z\}$, thus our queries are well-typed. Moreover, since $y \neq z$, at least one of $x \neq y$ and $x \neq z$ must be true. The corresponding query will yield *no*, and we thus answer correctly.

Now consider the case where $x \notin U$. Then $A_y = \{x, y\}$ and $A_z = \{x, z\}$. Our names for A_y and A_z thus can change arbitrarily late to become a name for $\{x\}$ instead. While the computations $A_y \in \{\{x\}\}$? and $A_z \in \{\{x\}\}$? may be ill-typed, they must not produce output inconsistent with returning a positive answer for $\{x\} \in \{\{x\}\}$?. Thus, we will not receive a *no*-answer, and hence not answer incorrectly. ☐

Corollary 15. *Let* $\mathbf{X}$ *be computably Hausdorff, computably admissible and contain two computable points. Then* $\mathbf{X} \cong \mathbf{X}^{\mathsf{dd}}$.

Proof. Computability of $\mathrm{id} : \mathbf{X} \to \mathbf{X}^{\mathsf{dd}}$ is available without assumptions (Corollary 5). For the converse direction, note that given $\{\{x\}\} \in \mathbf{X}^{\mathsf{dd}}$ we can first invoke Proposition 11 (since $\mathbf{X}$ is assumed to be computably Hausdorff) to obtain $\{x\} \in \mathbf{X}^{\mathsf{d}}$. We then use Lemma 14 to get $x \in \mathbf{X}$. Note that since $\mathbf{X}$ is computably Hausdorff, having two computable points in $\mathbf{X}$ yields two computable points in $\mathbf{X}^{\mathsf{d}}$ by Observation 6. ☐

Note that combining Observation 6 and Corollaries 9 and 15 yields the effectivization of [7, Example 4.2].

Proposition 16. *If* $\mathbf{X}^{\mathsf{d}}$ *contains two computable points and is computably Hausdorff, then* $\mathbf{X}$ *is computably compact.*

Proof. Let $\{a\}, \{b\} \in \mathbf{X}^{\mathsf{d}}$, $a \neq b$, be the computable points available to us. To show that $\mathbf{X}$ is computably compact, we show that we can, given $A \in \mathcal{A}(\mathbf{X})$, recognize if $A = \emptyset$. That $\mathbf{X}^{\mathsf{d}}$ is computably Hausdorff means we have available to us a computable map $\mathrm{isNotEqual} : \mathbf{X}^{\mathsf{d}} \times \mathbf{X}^{\mathsf{d}} \to \mathbb{S}$. We attempt the computation $\mathrm{isNotEqual}(\{a\} \cup A, \{b\} \cup A)$, and claim that the answer to this correctly identifies whether $A = \emptyset$.

If $A = \emptyset$, we are computing $\mathrm{isNotEqual}(\{a\}, \{b\})$, which has to answer *yes*. If $A \neq \emptyset$, then $\mathrm{isNotEqual}(\{a\} \cup A, \{b\} \cup A)$ is not well-typed. Consider some $c \in A$. Then names for both $\{a\} \cup A$ and $\{b\} \cup A$ can change arbitrarily late to be a name for $\{c\}$ instead. Thus, the computation of $\mathrm{isNotEqual}(\{a\} \cup A, \{b\} \cup A)$ must never output anything that would be inconsistent with $\mathrm{isNotEqual}(\{c\}, \{c\})$, i.e., it must never answer *yes*. We thus obtain the desired behaviour. ☐

Iterated duality. The following observation is straightforward for T_1-spaces, but false in general (see Example 27 below).

Observation 17 *If $f : \mathbf{X} \to \mathbf{Y}$ is a computable bijection, then $f^{-1} : \mathbf{Y}^{\mathsf{d}} \to \mathbf{X}^{\mathsf{d}}$ is well-defined and computable.*

For the topological de Groot dual, Kovár has shown that taking iterated duals will yield at most four distinct topological spaces [7]. For T_1 represented spaces, the iterated dual will only yield at most three distinct represented spaces, with an argument that is similar to but simpler than the one by Kovár. We will see later an example showing that $\mathbf{X}, \mathbf{X}^{\mathsf{d}}$ and $\mathbf{X}^{\mathsf{dd}}$ can indeed be three non-isomorphic represented spaces (Section 4).

Corollary 18. $\mathbf{X}^{\mathsf{d}} \cong \mathbf{X}^{\mathsf{ddd}}$.

Proof. That $\mathrm{id} : \mathbf{X}^{\mathsf{d}} \to \mathbf{X}^{\mathsf{ddd}}$ is computable is just a consequence of Corollary 5. To get the computability of $\mathrm{id} : \mathbf{X}^{\mathsf{ddd}} \to \mathbf{X}^{\mathsf{d}}$, we apply Observation 17 to $\mathrm{id} : \mathbf{X} \to \mathbf{X}^{\mathsf{dd}}$ from Corollary 5. □

Corollary 19. *Let $\mathbf{X}$ contain two computable points. Then $\mathbf{X}^{\mathsf{dd}}$ is computably Hausdorff iff $\mathbf{X}$ is.*

Proof. If $\mathbf{X}$ is computably Hausdorff, so is its admissibilification $\mathbf{X}_\kappa$, and they have the same dual. Corollary 15 then yields $\mathbf{X}_\kappa \cong \mathbf{X}^{\mathsf{dd}}$, so the latter is computably Hausdorff.

Conversely, if $\mathbf{X}^{\mathsf{dd}}$ is computably Hausdorff, then by Corollary 12, $\mathbf{X}^{\mathsf{ddd}}$ is computably compact (we can lift a computable point from $\mathbf{X}$ to $\mathbf{X}^{\mathsf{dd}}$ by Corollary 5). Since $\mathbf{X}^{\mathsf{d}} \cong \mathbf{X}^{\mathsf{ddd}}$ by Corollary 18, $\mathbf{X}^{\mathsf{d}}$ is computably compact. Then Corollary 13 shows that $\mathbf{X}$ is computably Hausdorff. □

Let us confirm that the above completes the proof of Theorem 3. The item (1) follows from Corollary 5 (2). The item (2) follows from Corollary 18. For the item (3), (a)→(b): Corollary 15. (b)→(c): trivial. (a)↔(c): Corollary 19. (a)↔(d): Corollaries 12 and 13. (a)↔(e): Observation 6. (a)→(f): Proposition 11. (f)→(e): Theorem 3 (1). For the item (4), (a)↔(c): Observation 7 (2) and Proposition 16. (b)↔(c)↔(e): Apply Theorem 3 (3) (d)↔(a)↔(e) to $\mathbf{X}^{\mathsf{d}}$. (a)↔(d): Observation 7 (1) and Corollary 9. The item (5) follows from Corollary 10.

3 Duality for non-T_1 represented spaces

Notation. We now leave behind the tacit restriction to T_1-spaces. We recap some basic notions we will need for our discussion here. For a represented space $\mathbf{X}$, we shall write $\leq_{\mathbf{X}}$ for its specialization preorder; which is defined as $x \leq_{\mathbf{X}} y$ iff every open containing x also contains y. Equivalently, $\overline{\{x\}} \subseteq \overline{\{y\}}$. A set $A \subseteq X$ is *saturated* if it is an intersection of open sets. The saturation of a set A is $\uparrow A := \bigcap_{\{U \in \mathcal{O}(\mathbf{X}) \mid A \subseteq U\}} U$. Note that the saturation of a compact set is also compact since an open cover of A always covers the saturation $\uparrow A$. If we consider only singletons, the topological closure corresponds to the $\leq_{\mathbf{X}}$-downward closure, and the saturation corresponds to the $\leq_{\mathbf{X}}$-upward closure.

One can see that the de Groot dual inverts the specialization preorder; that is, $x \leq_{\mathbf{X}^{\mathsf{d}}} y$ iff $y \leq_{\mathbf{X}} x$. In particular, $\mathbf{X}$ is T_1 iff $\mathbf{X}^{\mathsf{d}}$ is T_1. Thus, the sequences of iterated duals of T_1 and non-T_1 spaces never intersect. Recall that, for T_0-case, the map $x \mapsto \overline{\{x\}} = \downarrow x$ is bijective, so one can think of an underlying set of $\mathbf{X}^{\mathsf{d}}$ as X by identifying $\overline{\{x\}}$ with x. Hereafter, we assume that a represented space $\mathbf{X}$ is always T_0.

Iterated duality for non-T_1 represented spaces. Below we observe that the iteration sequence of the de Groot dual of a represented space terminates in at most three steps, even if we start from a non-T_1 space. This contrasts with the existence of a topological space whose iterated dual sequence does not terminate in at three steps [7].

Theorem 20. $\mathbf{X}^{\mathsf{ddd}} \simeq \mathbf{X}^{\mathsf{d}}$ *for any represented T_0-space* $\mathbf{X}$.

To see this, we first see the following analogue of Observation 4.

Observation 21 *The map* $\not\leq_{\mathbf{X}} : \mathbf{X} \times \mathbf{X}^{\mathsf{d}} \to \mathbb{S}$ *is computable.*

Proof. As $\mathcal{A}(\mathbf{X}) \simeq \mathbb{S}^{\mathbf{X}}$, the non-membership relation $\notin : \mathbf{X} \times \mathcal{A}(\mathbf{X}) \to \mathbb{S}$ is exactly the evaluation map, so it is computable. For $x, y \in X$, note that $x \notin \overline{\{y\}}$ iff $x \not\leq_{\mathbf{X}} y$. Thus, the non-membership relation $\notin$ restricted to $\mathbf{X} \times \mathbf{X}^{\mathsf{d}}$ is exactly the relation $\not\leq_{\mathbf{X}}$ via the identification of $\overline{\{y\}}$ with $y \in \mathbf{X}^{\mathsf{d}}$. Therefore, $\not\leq_{\mathbf{X}} : \mathbf{X} \times \mathbf{X}^{\mathsf{d}} \to \mathbb{S}$ is computable. □

We see that Corollary 5 (2) also holds for non-T_1 spaces.

Corollary 22. $\mathrm{id} : \mathbf{X} \to \mathbf{X}^{\mathsf{dd}}$ *is computable.*

Proof. Currying the function $\not\leq_{\mathbf{X}}$ in Observation 21 yields the saturation $x \mapsto \uparrow_{\mathbf{X}}\{x\} = \{y \in X : x \leq_{\mathbf{X}} y\} : \mathbf{X} \to \mathcal{A}(\mathbf{X}^{\mathsf{d}})$. Note that an element of $\mathbf{X}^{\mathsf{dd}}$ is of the form $\downarrow_{\mathbf{X}^{\mathsf{d}}}\{x\}$, which turns out to be $\uparrow_{\mathbf{X}}\{x\}$ since the de Groot dual inverts the specialization preorder as mentioned above. Hence, the range of the saturation is exactly $\mathbf{X}^{\mathsf{dd}}$, so $\mathrm{id} : \mathbf{X} \to \mathbf{X}^{\mathsf{dd}}$ is computable. □

In general, for (non-T_1) topologies σ and τ (on the same underlying set), the condition $\sigma \subseteq \tau$ does not imply $\tau^{\mathsf{d}} \subseteq \sigma^{\mathsf{d}}$, but if σ and τ have the same specialization order, this does hold. Based on this observation, we see the following non-T_1 analogue of Observation 17.

Observation 23 *If* $f : \mathbf{X} \to \mathbf{Y}$ *is computable, and* $f : (X, \leq_{\mathbf{X}}) \to (Y, \leq_{\mathbf{Y}})$ *is an order isomorphism, then* $f^{-1} : \mathbf{Y}^{\mathsf{d}} \to \mathbf{X}^{\mathsf{d}}$ *is well-defined and computable.*

Proof. Computability of $f : \mathbf{X} \to \mathbf{Y}$ implies computability of $f^{-1} : \mathcal{A}(\mathbf{Y}) \to \mathcal{A}(\mathbf{X})$ (via computability of $\mathbf{Y}$). For any $y \in Y$, by surjectivity, we have some $x \in X$ such that $f(x) = y$. Since f is an order isomorphism, $x' \leq_{\mathbf{X}} x$ if and only if $f(x') \leq_{\mathbf{Y}} f(x) = y$. This implies that $f^{-1}[\downarrow_{\mathbf{Y}}\{y\}] = \downarrow_{\mathbf{X}}\{x\}$. Hence, $f^{-1} : \mathbf{Y}^{\mathsf{d}} \to \mathbf{X}^{\mathsf{d}}$ is well-defined and computable. □

Proof (Theorem 20). That $\mathrm{id}\colon \mathbf{X}^{\mathsf{d}} \to \mathbf{X}^{\mathsf{ddd}}$ is computable is just a consequence of Corollary 22. To get the computability of $\mathrm{id}\colon \mathbf{X}^{\mathsf{ddd}} \to \mathbf{X}^{\mathsf{d}}$, note that $\mathbf{X}$ and $\mathbf{X}^{\mathsf{dd}}$ have the same specialization order since the de Groot dual inverts the specialization preorder as mentioned above. In particular, $\mathrm{id}\colon \mathbf{X} \to \mathbf{X}^{\mathsf{dd}}$ is an order isomorphism. Hence, we just need to apply Observation 23 to $\mathrm{id}\colon \mathbf{X} \to \mathbf{X}^{\mathsf{dd}}$ from Corollary 22. □

4 Examples

The cofinite topology on $\mathbb{N}$. An important example to illustrate the duality between Hausdorff spaces and compact T_1-spaces is the observation that $\mathbb{N}^{\mathsf{d}} = \mathbb{N}_{\mathrm{cof}}$, where $\mathbb{N}_{\mathrm{cof}}$ are the natural numbers equipped with the cofinite topology. We then also have that $(\mathbb{N}_{\mathrm{cof}})^{\mathsf{d}} = \mathbb{N}$.

The cocylinder topology on Baire space. As announced in Section 2, we give an example where $\mathbf{X} \simeq \mathbf{X}^{\mathsf{dd}}$ is not necessarily true even if $\mathbf{X}$ is computably compact and T_1.

Definition 24. *The cocylinder topology τ_c on $\mathbb{N}^{\mathbb{N}}$ is generated by co-cylinders $\{X : X \not\succ \sigma\}$ where σ ranges over finite strings. We write $\mathbb{N}^{\mathbb{N}}_c = (\mathbb{N}^{\mathbb{N}}, \tau_c)$.*

The space $\mathbb{N}^{\mathbb{N}}_c$ is second-countable, computably compact and T_1. It is neither Hausdorff nor sober (and thus not stably compact). We see below that $(\mathbb{N}^{\mathbb{N}}_c)^{\mathsf{d}} \simeq \mathbb{N}^{\mathbb{N}}$ and thus $(\mathbb{N}^{\mathbb{N}}_c)^{\mathsf{dd}} \simeq (\mathbb{N}^{\mathbb{N}})^{\mathsf{d}}$, but $(\mathbb{N}^{\mathbb{N}})^{\mathsf{d}}$ is not second-countable (see Section 5), so $(\mathbb{N}^{\mathbb{N}}_c)^{\mathsf{dd}} \not\simeq \mathbb{N}^{\mathbb{N}}_c$.

Proposition 25. $(\mathbb{N}^{\mathbb{N}}_c)^{\mathsf{d}} \simeq \mathbb{N}^{\mathbb{N}}$

Proof. First note that a name of $x \in \mathbb{N}^{\mathbb{N}}_c$ is an enumeration $(\sigma_n)_{n\in\mathbb{N}}$ of all non-prefixes of x. And, a name of a closed set $A \in \mathcal{A}(\mathbb{N}^{\mathbb{N}}_c)$ is a sequence $D = (D_n)_{n\in\mathbb{N}}$ of finite sets D_n of strings such that $x \in A$ iff, for any $n \in \mathbb{N}$, D_n contains a prefix of x. Thus, given an $\mathbb{N}^{\mathbb{N}}$-name of x, by putting D_n to be the singleton $\{x \upharpoonright n\}$, where $x \upharpoonright n$ is the prefix of x of length n, we get a name of $\{x\} \in \mathcal{A}(\mathbb{N}^{\mathbb{N}}_c)$. This shows that $\mathrm{id}\colon \mathbb{N}^{\mathbb{N}} \to (\mathbb{N}^{\mathbb{N}}_c)^{\mathsf{d}}$ is computable.

Conversely, assume that a name D of a closed set $A \in \mathcal{A}(\mathbb{N}^{\mathbb{N}}_c)$ is given. From such a sequence D, one may construct a finite-branching tree whose infinite paths correspond to the elements of A. To see this, we inductively construct a sequence $E = (E_n)_{n\in\mathbb{N}}$ of finite sets of strings as follows: Let E_0 be the singleton consisting of the empty string. Assume that E_n has already been constructed. For each $\sigma \in E_n$, and each $\tau \in D_n$ which is comparable with σ, put the longer of σ and τ into E_{n+1}. By leaving only shorter strings in E_{n+1}, we may assume that elements of E_{n+1} are pairwise incomparable. Note that E_{n+1} is contained in the upward closure of E_n (w.r.t. the prefix order). We claim that $x \in A$ iff E_n has a prefix of x for any $n \in \mathbb{N}$. For the backward direction, note that if E_{n+1} has a prefix of x then so does D_n. For the forward direction, if $x \in A$, one can inductively ensure that E_n contains a prefix of x. By the assumption $x \in A$, D_n also contains a prefix of x, so a prefix of x survives in E_{n+1}.

Now, the downward closure of $\bigcup_{n\in\mathbb{N}} E_n$ yields a finite-branching tree T_E. If A is a singleton $\{x\}$, by the above arguments, one can see that T_E has a unique infinite path x. However, we only have an enumeration of the tree T_E which is not pruned, so it is not straightforward to compute an $\mathbb{N}^\mathbb{N}$-name of the unique path x. To overcome this difficulty, note that only one of the elements of E_n is a prefix of x. If $\sigma \in E_n$ is not a prefix of x, we claim that there exists $m > n$ such that E_m fails to have an extension of σ. Otherwise, for any $m > n$, E_m has an extension τ_m of σ. If $(\tau_m)_{m>n}$ is eventually constant, say τ, then almost all D_m contain an initial segment of τ, so any infinite string extending τ must be a path through T_E, which is impossible. Hence, $(\tau_m)_{m\in\mathbb{N}}$ contains infinitely many different strings in T_E extending σ. Since T_E is finite-branching, König's lemma implies that T_E has an infinite path extending σ, which is again impossible by our assumption. This verifies the claim, which shows that $\sigma \in E_n$ not being a prefix of x is semidecidable. Wait for all but one string in E_n to turn out not to be a prefix of x. Then the last remaining one turns out to be a prefix of x. In this way, we can compute a $\mathbb{N}^\mathbb{N}$-name of the unique path x, which shows that $\mathrm{id}\colon (\mathbb{N}^\mathbb{N}_c)^\mathsf{d} \to \mathbb{N}^\mathbb{N}$ is computable. □

The lower reals. The following example shows that we need to distinguish a space being isomorphic to its dual and being equal to its dual: The lower reals and the upper reals are isomorphic (with $x \mapsto -x$ being a computable isomorphism), but not equal (as $\mathrm{id} : \mathbb{R}_< \to \mathbb{R}_>$ is not computable).

Proposition 26. $\mathbb{R}_<^\mathsf{d} = \mathbb{R}_>$

The following shows that Observation 17 (about being able to reverse the direction of a computable bijection by taking the dual) does not hold once we move beyond T_1-spaces:

Example 27. $\mathrm{id} : \mathbb{R} \to \mathbb{R}_<$ is a computable bijection, yet $\mathrm{id} : \mathbb{R}_<^\mathsf{d} \to \mathbb{R}^\mathsf{d}$ is not computable.

5 The Point Degree Spectrum of $(\mathbb{N}^\mathbb{N})^\mathsf{d}$

As an application for de Groot duality, we show that, relative to any oracle, the point degree spectrum of the de Groot dual of $\mathbb{N}^\mathbb{N}$ contains non-enumeration degrees. The point degree spectrum links the study of recursion-theoretic degree structures such as the Medvedev degrees, enumeration degrees and Turing degrees to σ-homeomorphism types of topological spaces [5, 6, 10].

Let $\mathbf{X}$ and $\mathbf{Y}$ be represented spaces. For $x \in \mathbf{X}$ and $y \in \mathbf{Y}$ we write $y \leq_M x$ if there exists a partial computable function $F\colon\subseteq \mathbf{X} \to \mathbf{Y}$ such that $F(x) = y$; that is, given a name of x, one can effectively find a name of y.

Definition 28. *A non-computable point $x \in \mathbf{X}$ is* $\mathbb{S}^\mathbb{N}$-quasi-minimal *if for any $y \in \mathbb{S}^\mathbb{N}$, $y \leq_M x$ implies that y is computable.*

As $\mathbb{S}^\mathbb{N}$ is a universal second-countable T_0 space, we find that a $\mathbb{S}^\mathbb{N}$-quasi-minimal point is $\mathbf{Y}$-quasi-minimal for any second countable space $\mathbf{Y}$. The degrees of points in $\mathbb{S}^\mathbb{N}$ are exactly the enumeration degrees, so another perspective on $\mathbb{S}^\mathbb{N}$-quasi-minimal points is that they are non-computable points not computing any non-trivial enumeration degree.

Theorem 29. *Relative to any oracle, there are continuum many $\mathbb{S}^\mathbb{N}$-quasi-minimal $(\mathbb{N}^\mathbb{N})^\mathsf{d}$-degrees.*

In the following, we write $x^\mathbf{X}$ to emphasize that x is a point in the represented space $\mathbf{X}$. To avert superscript-overload, we will write $\mathcal{E}$ for $\mathbb{S}^\mathbb{N}$ and $\mathcal{B}$ for $\mathbb{N}^\mathbb{N}$.

Lemma 30. *If $y^\mathcal{E} \leq_\mathrm{T} x^{\mathcal{B}^\mathsf{d}}$, then one of the following must hold:*

1. *$y^\mathcal{E}$ is computable.*
2. *$x^\mathcal{B} \leq_\mathrm{T} x^{\mathcal{B}^\mathsf{d}} \oplus (\mathbb{N} \setminus y)^\mathcal{E}$*

Proof (Theorem 29). Given an oracle z, it is easy to construct a $\Pi^0_1(z)$ singleton $\{x\}$ in $\mathbb{N}^\mathbb{N}$ such that $x \not\leq_\mathrm{T} z'$ (see [8, Exercises XII.2.14 (d), and XII.2.15 (e)]). Moreover, if $\{x\}$ is such a $\Pi^0_1(z)$, then so is $\{x \oplus z\}$. We will write $x_z := x \oplus z$ where x is constructed from z in this manner.

Now, given an oracle r, consider any $z \geq_T \mathcal{O}^r$, where $\mathcal{O}^r$ is the hyperjump of r, that is, a $\Pi^1_1(r)$-complete subset of $\mathbb{N}$. Then, $\{x_z\}$ is not a $\Pi^0_1(r)$ singleton; otherwise, x_z is Δ^1_1 in r [8, Proposition XII.2.16], and thus $x_z \leq_T \mathcal{O}^r \leq_T z$, a contradiction.

We will show that $(x_z)^{\mathcal{B}^\mathsf{d}}$ is $\mathcal{E}$-quasiminimal relative to r. We argued above that $(x_z)^{\mathcal{B}^\mathsf{d}}$ is not computable relative to r. Assume that some non-computable $y \in \mathcal{E}$ satisfies $y^\mathcal{E} \leq_\mathrm{T} (x_z)^{\mathcal{B}^\mathsf{d}}$ relative to r. Then it follows by Lemma 30 that $x_z^\mathcal{B} \leq_\mathrm{T} x_z^{\mathcal{B}^\mathsf{d}} \oplus (\mathbb{N} \setminus y)^\mathcal{E} \oplus r$. We know that z can compute $x_z^{\mathcal{B}^\mathsf{d}}$ and r, and thus also $y^\mathcal{E}$. But then z' computes z and $(\mathbb{N} \setminus y)^\mathcal{E}$, hence $x_z \leq_\mathrm{T} z'$. But we constructed x_z such that $x \not\leq_\mathrm{T} z'$, and thus have reached a contradiction. It follows that $(x_z)^{\mathcal{B}^\mathsf{d}}$ is $\mathcal{E}$-quasiminimal relative to r. As there are continuum many $z \geq_T \mathcal{O}^r$, and since $z_1 \neq z_2$ implies $x_{z_1} \neq x_{z_2}$, the claim follows. □

Having continuum many $\mathbb{S}^\mathbb{N}$-quasi-minimal points has a topological interpretation:

Corollary 31. *For any second-countable T_0 space $\mathcal{Y}$, if $f : (\mathbb{N}^\mathbb{N})^\mathsf{d} \to \mathcal{Y}$ can be decomposed into countably many continuous functions, then there is a continuum-sized set A such that the image $f[A]$ is countable.*

Proof. Let $f : \mathcal{B}^\mathsf{d} \to \mathcal{Y}$ be a σ-continuous function, where $\mathcal{Y}$ is a second-countable T_0 space. Then, via an embedding $\mathcal{Y} \hookrightarrow \mathbb{S}^\mathbb{N}$, one can think of f as a σ-continuous function $f : \mathcal{B}^\mathsf{d} \to \mathbb{S}^\mathbb{N}$. Then, f is σ-computable relative to some oracle r. Note that $f(x) \leq_M x \oplus r$. Let $A \subseteq \mathcal{B}^\mathsf{d}$ be the set of all points which are second-countable quasi-minimal relative to r, that is, if $x \in A$ then $f(x)$ is r-computable. Then, since there are only countably many r-computable points in $\mathbb{S}^\mathbb{N}$, the range of $f[A]$ is countable as desired. □

The idea of the proof of Theorem 29 is to exploit the difference in computability theoretic strength between explicit and implicit definability. A similar idea has been used in the classical theory of implicit definability (Π^0_1 singletons), so we mention its historical origin to conclude the discussion.

Recall that an object is implicitly definable (in arithmetic) if it is a unique solution of an (arithmetical) predicate; see e.g. Odifreddi [8, Definition XII.2.13]. One of the triggers that made this notion worth studying in logic was, for example, the following observation: Tarski's truth undefinability theorem tells us that arithmetical truth is not explicitly definable in arithmetic; nevertheless, arithmetical truth is known to be implicitly definable in arithmetic. What the latter means is that arithmetical truth is a unique solution of an arithmetical predicate; more precisely, the set of codes of true sentences in first order arithmetic is an arithmetical singleton in $\mathcal{P}(\mathbb{N})$ (see e.g. Odifreddi [8, Definition XII.2.13 and Proposition XII.2.19]).

In this way, implicit definability can often encode powerful information, and in fact, we have used this property to analyze the point degree spectrum of the de Groot dual of $\mathbb{N}^{\mathbb{N}}$.

Acknowledgements

We are grateful to Matthew de Brecht for fruitful discussions. We are also grateful to the anonymous referees for valuable suggestions and comments.

References

1. Brattka, V., de Brecht, M., Pauly, A.: Closed choice and a uniform low basis theorem. Ann. Pure Appl. Logic **163**(8), 986–1008 (2012)
2. Escardó, M.: Synthetic topology: of data types and classical spaces. Electronic Notes in Theoretical Computer Science **87**, 21–156 (2004)
3. Goubault-Larrecq, J.: Non-Hausdorff topology and domain theory: Selected topics in point-set topology, New Mathematical Monographs, vol. 22. Cambridge University Press, Cambridge (2013)
4. de Groot, J., Herrlich, H., Strecker, G.E., Wattel, E.: Compactness as an operator. Compositio Math. **21**, 349–375 (1969)
5. Kihara, T., Ng, K.M., Pauly, A.: Enumeration degrees and non-metrizable topology. arXiv:1904.04107 (2019)
6. Kihara, T., Pauly, A.: Point degree spectra of represented spaces. Forum Math. Sigma **10**, Paper No. e31, 27 (2022)
7. Kovár, M.M.: At most 4 topologies can arise from iterating the de Groot dual. Topology Appl. **130**(2), 175–182 (2003)
8. Odifreddi, P.G.: Classical Recursion Theory. Vol. II, Studies in Logic and the Foundations of Mathematics, vol. 143. North-Holland Publishing Co., Amsterdam (1999)
9. Pauly, A.: On the topological aspects of the theory of represented spaces. Computability **5**(2), 159–180 (2016)
10. Pauly, A.: Enumeration degrees and topology. In: Sailing routes in the world of computation, Lecture Notes in Comput. Sci., vol. 10936, pp. 328–337. Springer, Cham (2018)