COMPARING PRESENTATIONS OF COMPACT SPACES

NOAM GREENBERG, ALEXANDER MELNIKOV, ISABELLA SCOTT,
AND DANIEL TURETSKY

Abstract. For computable Polish spaces A, B, the relation "A is computably homeomorphic to B" is a pre-order rather than an equivalence relation. We investigate the *computable topological degrees* induced by this pre-order. In this paper, we restrict ourselves to Stone spaces. We prove several characterisation-type results in which an order-theoretic property is fully captured by some topological property of the space. We also investigate the degrees of 2^ω in more detail and apply classical results about Turing and enumeration degrees to draw conclusions about the computable topological degrees of this space.

Contents

1. Introduction 1
2. Preliminaries 7
3. Degrees of 2^ω 9
4. The 1-atom 21
5. Proof of Theorems 1.5 and 1.7 31
6. Upper bounds. Proof of Theorem 1.8 34
References 38

1. Introduction

We develop a systematic framework for computable compact metric spaces viewed up to computable homeomorphism. The framework is inspired by the study of computable dimension in computable structure theory. As we will see, the new framework leads to a pre-order and a degree structure rather than an equivalence relation. We test the new approach on the class of Stone spaces, for which we obtain a number of characterisations and structural results. But clearly, the framework is not restricted to this class.

1.1. **Computable dimension.** In computable structure theory, the notion of auto-dimension (computable dimension) has played a significant role in the past few decades. The idea is as follows. In [35], Mal'cev suggested that *computable presentations* of algebraic structures (such as groups, fields, and Boolean algebras) should be viewed up to *computable* isomorphism. Of course, the distinction between computable and abstract isomorphism occurs only for structures that are not finitely generated. However, in general, the relation of being computably isomorphic properly refines the isomorphism relation. Indeed, for any $\alpha \in \omega \cup \{\omega\}$, there exists an algebraic structure having exactly α computable presentations, up to computable isomorphism [14, 15, 13]; such an α is called the auto-dimension (the computable dimension) of the structure. Over the past 50 years, there has been a significant amount of research focused on the auto-dimension of algebraic structures; see, e.g., [46, 9, 17, 19] and the more recent works [43, 10, 52, 24, 33]. This topic is notorious for its technical depth, as well as for its counterintuitive strategies and results.

In computable topology and metric space theory, the situation becomes even more complex. In contrast with discrete algebra, even complete separable metric (Polish) spaces admit more than one natural notion of isomorphism, let alone computable isomorphism. Indeed, it is natural to consider metric spaces under isometry, quasi-isometry, bi-Lipschitz maps, homeomorphism, or homotopy, among other potential notions of similarity. Rather significant progress has been made only in the study of the computable auto-dimension of Polish and separable Banach spaces up to isometry; we cite [49, 41, 38, 36, 26]. In such studies, the intuition from computable algebra can sometimes be adapted to the separable setting, noting that the inverse of a computable isometry between computable Polish spaces is also necessarily computable. In particular, the notion of (isometric) auto-dimension still makes sense, and can also be any $\alpha \in \omega \cup \{\omega\}$ (as noted in [21]).

A lot less is known about computable presentability of Polish spaces viewed up to homeomorphism. Rather significant progress has been made only recently in a series of papers [6, 3, 23, 25, 34, 1, 30, 48], with some key results obtained independently by different researchers. However, virtually nothing is known about spaces viewed up to *computable* homeomorphism; to our best knowledge, the only works in this direction are [1, 37, 39, 42], but [39, 42] were concerned with computable Polish groups in which the group operation was also assumed to be computable.

Before we give the main definition of the paper and discuss the results, we need to review some classical notions from computable metric space theory.

1.2. **Computable and computably compact spaces.** The two most commonly used notions of computable presentability of a (separable) compact metric space are as follows:

(1) A *computable metric space* is given by a countable dense sequence (x_i) and an operator that computes, uniformly in i, j, the real number $d(x_i, x_j)$. If the space is complete we say it is a *computable Polish space*.
(2) A *computably compact space* is a (compact) computable metric space in which, additionally, we can list all finite (tuples of) of balls of the form $B(x_i, r)$, where $r \in \mathbb{Q}$, that cover the space.

We remark that all our spaces will be compact, and therefore complete. Thus, in (1), all metric spaces are Polish. We shall omit "metric" or "Polish" and say simply "computable space". The first of the two definitions is usually attributed to Ceitin [8] and Moschovakis [45]. The second definition is due to Mori, Tsujii, and Yasugi [44], but in many natural cases it is equivalent to the old notion of a computable closed set that can be traced back to Brouwer (see surveys [12, 27, 7]).

It is known that there exist examples of computable compact spaces which are not homeomorphic to any computably compact space [25, 34, 30, 12], and such spaces can be found in natural broad classes such as, e.g., solenoids [40, 34]. In contrast, among the totally disconnected compact spaces (the Stone spaces), every computable space is homeomorphic to a computably compact space [23, 12]. The proof of the latter positive result is, however, non-uniform, and the homeomorphism that can be extracted from the proof is non-effective in general.

One naturally wonders when computable metric spaces are *computably* homeomorphic to a computably compact one. As we will see, answering this question even in the special class of Stone spaces requires quite a bit of work.

1.3. **Computable maps between spaces.** The formal definition of a computable map $f : N \to M$ between computable spaces N and M will be given in the preliminaries. Informally, this means that f uniformly turns fast converging sequences into fast converging sequences. Equivalently, the map is effectively continuous, i.e., the open set $f^{-1}(B)$ can be effectively listed, for every basic open ball $B \subseteq M$. In particular, every computable map is continuous. For example, if $f : N \to M$ is a computable bijection, and M, N are compact, then f has to be a homeomorphism—recall that the inverse of a continuous bijection is also continuous in this special case. However, the crucial observation here is as follows:

The inverse of a computable bijection

$$f : N \to M$$

does *not* have to be computable.

We will illustrate this phenomenon with a couple of examples shortly. It follows that, for computable (compact) spaces,

"N is computably homeomorphic to M"

is a pre-order rather than an equivalence relation, and this leads to a *reducibility* between presentations.

1.4. **Computable topological reducibility.** Let M and N be computable Polish presentations of a compact metric space X.

Definition 1.1. We say that M is *computably topologically reducible* to N, written $M \leqslant_{ct} N$, if there is a computable bijection $f : N \to M$.

We will explain shortly why we chose $f : N \to M$ over $f : M \to N$ in the definition above; this is, of course, a matter of taste. It is clear that $\leqslant_{ct}$ is transitive and reflexive. It induces the *computable topological equivalence* relation $\equiv_{ct}$. In the notation of Definition 1.1, f^{-1} has to also be continuous (by compactness), and therefore $M \leqslant_{ct} N$ implies that M and N have to be homeomorphic. Thus, the relation is particularly well-behaved and well-suited for compact spaces, but one can of course drop the assumption of compactness and require f to be a homeomorphism.

Definition 1.2. Fix a compact space X and let $Comp(X)$ be the collection of all computable spaces homeomorphic to X. The *computable topological degrees* of X is the partial order

$$\mathbf{CT}(X) = (Comp(X), \leqslant_{ct}) / \equiv_{ct},$$

where $\leqslant_{ct}$ is identified with the induced partial order on equivalence classes modulo $\equiv_{ct}$.

Recall also that we could have defined the reduction symmetrically, i.e., by saying that M is below N if there is a computable homeomorphic map from M onto N (not from N onto M, as we did). However, we believe that our choice appears to be more natural. Indeed, it is not difficult to see that "being computably compact" is a *ct*-degree-invariant property, and furthermore the *ct*-degrees of computably compact spaces are *minimal* (Lemmas 2.7 and 2.8). Since computably compact presentations are "more effective", so one would expect them to be minimal rather than maximal.

We now give two examples illustrating what can "go wrong" with f^{-1} for a computable f.

Example 1.3. Let $B = [0, \Omega]$ and $A = [0, 1]$, where Ω is Chaitin's Omega, or any other positive left-c.e. real that is not computable. The metric is Euclidean, and the dense sets are rational. We claim that there is a computable homeomorphism $f : B \to A$ but no computable homeomorphism from A to B. Thus, $A <_{ct} B$.

To build a computable $f : [0, \Omega] \to [0, 1]$, we implement the following idea. We shall declare the pre-image of the interval $(1 - 2^{-s}, 1]$ (which is of course open in $[0, 1]$) to be $(\Omega_s, \infty) \cap [0, \Omega]$. As more points appear in the space $[0, \Omega]$, we will make progress in enumerating the open set $(\Omega_s, \infty) \cap [0, \Omega]$; this makes the set effectively open. The pre-images of open sub-intervals will be defined "naturally" elsewhere, up to some scaling. We omit further details.

However, the space $[0, \Omega]$ is evidently not computably compact, since this would make its diameter a computable real. If there were a computable $g : [0, 1] \to [0, \Omega]$, it would make $[0, \Omega]$ computably compact (Lemma 2.7). We conclude that

$$[0, 1] <_{ct} [0, \Omega].$$

Example 1.4. We identify the Cantor space with its natural computably compact presentation 2^ω. Of course, we can have a complete tree T with $[T] \cong 2^\omega$ where the branching at level n is controlled by a computable function $f(n, s)$ monotonic in the second argument. (We may assume $f(n, 0) = 2$.) Clearly, the collection of infinite paths through T, $[T]$, is computably compact iff we can compute $g(n) = \lim_s f(n, s)$. In particular, such a presentation does not have to be computably compact, and in this case $[T] \nleq_{tc} 2^\omega$, as in the previous example. We claim that, indeed, $[T] >_{tc} 2^\omega$. To see why, define a computable bijective $h : [T] \to 2^\omega$ in stages. If T expands at level n due to $g(n)[s] = f(n, s)$ increasing at stage s, we will declare the newly revealed part to be in the pre-image of one of the basic open sets. With some care and using careful choice of images of the newly enumerated parts of $[T]$, this idea can be implemented to build such an h.

Examples 1.3 and 1.4 provide further intuition: if we have a computable homeomorphic $f : B \to A$ then B can be thought of as being somewhat (topologically) bigger. In this case B may be reluctant to reveal important parts of itself for a while. In a typical diagonalisation scenario B will look exactly like A for a long time before revealing a hidden part X. Then we can often hide its scaled-down copy $f(X)$ inside of A; this may be counter-intuitive to an effective algebraist who'd rather expect A to be effectively embeddable into B in this scenario, resulting in a symmetrical definition.

1.5. **Results.** While it is easy to come up with examples similar to 1.3 and 1.4, to our surprise, proving deeper results about the computable topological degrees of even the most basic common spaces turned out to be a challenging task. We discovered that many standard techniques borrowed from other related topics consistently fail. For example, our reduction vaguely resembles the PR-reducibility in primitive recursive algebra (e.g., [28, 20, 2, 29, 31]), where the inverse of a primitive recursive isomorphism also does not have to be primitive recursive. However, this similarity too does not go beyond this observation, and neither the techniques nor the intuition can be transferred to our setting.

In this paper, we therefore restrict ourselves to Stone spaces, which are the totally disconnected compact metric spaces. These are known to recursion theorists as Π^0_1-classes, but our classes will not always be represented as paths through a computable tree. These spaces are dual to Boolean algebras under the well-known Stone duality. Our choice was dictated by our close familiarity with the class, and the abundance of techniques associated with Π^0_1-classes and computable Boolean algebras. Thus, we hoped that some of these techniques could be recycled to prove results about the *ct*-degrees. While some of these basic techniques were indeed useful, most of our proofs will rely on strategies and ideas which appear to be completely new. So, in the end, the familiarity with the class was perhaps the only real advantage.

Computable compactness and least degrees. Our first result addresses the question raised earlier: When is a computable space computably homeomorphic to a computably compact one? We prove:

Theorem 1.5. *For a computable Stone space S, the following conditions are equivalent:*

1. *Every ct-degree of S is above some computably compact degree.*
2. *The Cantor–Bendixson derivative $(S)'_{CB}$ of S is finite.*

In the theorem above, (1) says that every computable presentation of S can be computably homeomorphically transformed into a computably compact one. In particular, for such spaces, the minimal degrees are exactly the computably compact ones. However, we do not know the answer to the following question (in general, not only for Stone spaces):

Question 1.6. Is every minimal *ct*-degree necessarily computably compact?

Combined with some further known results, Theorem 1.5 implies:

Theorem 1.7. *For a (computable) Stone space S, TFAE:*

1. *$\mathbf{CT}(S)$ has a least degree;*
2. *S is finite.*

We wonder whether there exist infinite compact spaces having a least computable topological degree.

Common upper bound. It is always satisfying when a computability-theoretic property turns out to be equivalent to an algebraic or topological property. Theorem 1.7 is one such example. We discovered another quite natural property of *ct*-degrees that admits a nice algebraic characterisation.

Theorem 1.8. *Suppose $S = \widehat{B}$ is a Stone space with trivial perfect kernel. The following conditions are equivalent:*

1. *In $\mathbf{CT}(S)$, every pair of degrees has a common upper bound.*
2. *$(S)'_{CB}$ is finite.*

The characterisation from Theorem 1.8 holds for the broader class of Stone spaces that correspond to 1-decidably presentable atomic Boolean algebras (Theorem 6.3).

It appears that even for some rather familiar spaces, the structure of computable topological degrees can be rather complex. For example, in both Theorems 1.7 and 1.8, spaces with only finitely many limit points play a special role. We will see that for such seemingly elementary spaces, the techniques are quite unusual and subtle. Properties of such spaces will be thoroughly investigated in §4.

Another familiar space that will be important in many proofs is the Cantor space. The *ct*-degrees of the Cantor space also turned out to be far more intricate than we anticipated, so we decided to investigate its degrees specifically.

Case study: the Cantor space. The properties of $\mathbf{CT}(2^\omega)$ that follow from known and previously stated results are as follows:

(1) 2^ω has no least computable topological degree.
(2) There is a unique computably compact degree $\mathbf{0}_c \in \mathbf{CT}(2^\omega)$.

Property (1) is a special important case of Theorem 1.5 and will reappear (in a different form) as Theorem 3.1. We will clarify (2) in §3.1; it follows from results in [1]. Here, we only note that the unique computably compact degree of the Cantor space is the one that contains the 'natural' presentation of it by binary strings; we identify it with 2^ω.

Further (and certainly less elementary) facts about the Cantor space are summarised in the theorem below.

Theorem 1.9. *The following properties hold for* 2^ω*:*

(1) There exist $\mathbf{a}, \mathbf{b} \in \boldsymbol{CT}(2^\omega)$ *such that* $\mathbf{a} \mid \mathbf{b}$ *and* $\inf\{\mathbf{a}, \mathbf{b}\} = \mathbf{0}_c (= \deg(2^\omega))$.
(2) There exist two computable presentations A *and* B *of* 2^ω *such that* $A \equiv_{ct} B$, *but* A *and* B *are not bi-computably homeomorphic.*

In the theorem, (1) says that $\mathbf{a}$ and $\mathbf{b}$ form a "minimal pair" over 2^ω; however, 2^ω is not the least element (so maybe, more properly, 2^ω "braches" in $\mathbf{CT}(2^\omega)$). Part (2) says that the analogue of the Schröder–Bernstein Theorem *fails* for computable homeomorphisms, and indeed it fails already between computable copies of 2^ω. This illustrates the subtle distinction between $\equiv_{ct}$ and the relation "being bi-computably homeomorphic". Of course, to obtain such an example one has to specifically look for trouble; however, one perhaps would not expect such an example to exist for a homogeneous space such as 2^ω.

The proof of Theorem 1.9 relies on classical results from enumeration and Turing degrees, combined with careful coding techniques. We conjecture that, with some effort, both (1) and (2) of Theorem 1.9 can be pushed to characterisations similar to Theorems 1.5, 1.7, and 1.8. However, to keep the paper concise, we shall be satisfied with the special natural case of 2^ω. The reader may find it more intriguing that, even in the special case of 2^ω, we don't know at present whether there exists a minimal non-computably compact degree (cf. Question 1.6). We also don't know whether $\mathbf{CT}(2^\omega)$ has the property described in Theorem 1.8; we leave this open.

The rest of the paper consists of a brief preliminaries section and the proofs of the results stated in the introduction. The proofs are split into various special cases, the two most significant ones being 2^ω and $\mathcal{A}_1$ which is the compact space with exactly one limit point. This material is split into subsections whose titles should be sufficiently self-descriptive and easy to navigate. Some of the proofs in the earlier parts of the paper are presented in much more detail than some of the later proofs. This is done either when the later arguments are indeed very similar to the arguments presented earlier, or when we believe that the reader should have no problem reconstructing the routine formal details. However, we felt compelled to present more detailed proofs at least once for each type of argument.

2. Preliminaries

The paper assumes some familiarity with the class of (countable) Boolean algebras, Stone duality, and with the basic definitions associated with Boolean algebras and their dual spaces. The standard reference is [16], but our notation will be consistent with [11, Chapter 4] where all these basic notions can be found. In this section, we briefly go over the definitions related to computable metric space theory since notation and terminology in the literature is not quite as unified.

2.1. Computable Polish spaces.

Definition 2.1. A *computable* (presentation of a) *Polish space* is given by: (1) a dense sequence $(x_i)_{i\in\omega}$, perhaps with repetitions, and (2) a computable function f which, given $i, j, s \in \omega$, outputs $r = \dfrac{n}{m} \in \mathbb{Q}$ such that $|d(x_i, x_j) - r| < 2^{-s}$, where d is the metric on the space.

We view spaces up to homeomorphism, and thus we require that the metric is compatible with the topology. We also require that the metric is complete, and so $M = \overline{(x_i)_{i\in\omega}}$. This is not really a restriction in the compact case, since every compact metric space is necessarily complete. Assuming the space is infinite, we can always uniformly effectively remove repetitions from the dense sequence and assume $x_i \neq x_j$ whenever $i \neq j$ (folklore; e.g. [11, Exercise 2.4.32]).

Points x_j from this sequence are called *special, ideal,* or (less frequently) *rational.* A *basic open ball* is a ball of the form $B(x_j, r) = \{y \in M : d(x_j, y) < r\}$. Here, x_j is a special point and $r \in \mathbb{Q}$ is positive. We also always represent rational numbers as fractions when possible. In particular, a basic open ball is assumed to have its radius represented as a fraction.

We say that an open set V is c.e. (in a computable Polish space M) if V is a c.e. union of basic open balls represented in this way. Say that a sequence of special points $(y_j)_{j\in\omega}$ is *fast Cauchy* if $d(y_j, y_{j+1}) < 2^{-j}$, for all j. The *name* of a point $x \in M$ of a computable Polish space M is the set $N^x = \{B \ni x : B \text{ is basic open}\}$. A point is computable if it admits a c.e. name. (Equivalently, if there exists a computable fast Cauchy sequence converging to the point.)

2.2. Computably compact spaces.

Definition 2.2 ([44])**.** A computable Polish space M with a dense set $(x_i)_{i\in\omega}$ is *computably compact* if there is a computable function which, given n, outputs a finite tuple $i_0, \dots, i_k$ of natural numbers such that $M = B(x_{i_0}, 2^{-n}) \cup \dots \cup B(x_{i_k}, 2^{-n})$, i.e., it is a finite open 2^{-n}-cover of the space.

Proposition 2.3 (Folklore)**.** For a computable Polish space $M = \overline{(x_i)_{i\in\omega}}$, the following are equivalent:

(1) M is computably compact (Definition 2.2).
(2) For every n, one can effectively produce a finite cover of M by basic closed 2^{-n}-balls.
(3) There is a computably enumerable list of *all* finite open covers of the space by basic open balls.

(4) There is an effective procedure which, given an enumeration of a countable cover of the space by basic open balls, outputs a finite sub-cover.
(5) There is a computable function $h\colon \mathbb{N} \to \mathbb{N}$ such that $M = \bigcup_{i \leqslant h(n)} B(x_i, 2^{-n})$.

See [12] for a proof. It is also well-known that the diameter of a computably compact space is a computable real.

Notation 2.4. For a (compact) computable Polish space M, we write $FinCov(M)$ for the collection of all finite covers of the space by basic open balls. In $FinCov(M)$, each such cover is represented as a tuple of parameters describing the cover (the indices of the centres and the fractions representing the respective radii).

It is clear that a (compact) computable Polish space M is computably compact iff $FinCov(M)$ is c.e.

2.3. **Computable maps between spaces.** Fix two computable Polish spaces.

Definition 2.5. A map $f : X \to Y$ is *computable* if, for every basic open B in Y, the set $f^{-1}(B)$ is c.e. open uniformly in the description of B.

Since we will have to build some maps in stages, we will need a local definition of a computable map in terms of pairs of basic open balls.

Definition 2.6 (Effective continuity, the ε-δ version)**.** A function $f\colon X \to Y$ is *effectively continuous* if there is a c.e. family F of pairs (D, E) of (indices of) basic open balls such that

(C1) for every $(D, E) \in F$, we have $f(D) \subseteq E$;
(C2) for every $x \in X$ and every basic open $E \ni f(x)$, there exists a basic open D with $(D, E) \in F$ and $x \in D$.

It is well-known that this definition is equivalent to Definition 2.5. Equivalently, f is computable if it uniformly effectively turns fast Cauchy sequences (in X) into fast Cauchy sequences (in Y); a sequence $(x_i)_{i\in\omega}$ is fast Cauchy if $d(x_i, x_{i+1}) < 2^{-i}$.

Lemma 2.7 (Folklore)**.** *Suppose $f : B \to A$ is a surjective computable map between computable Polish spaces.*

(1) If B is computably compact, then so is A.
(2) If B is computably compact and f is also injective (thus, a homeomorphism), then f^{-1} is also computable.

A slightly more general version of (1) of the lemma will appear as Lemma 3.13 later. The proof of (2) can be found in, e.g., [12, 11].

From Lemma 2.7, we immediately obtain:

Lemma 2.8. *Suppose B is a computably compact presentation of M. Then the ct-degree of B is minimal.*

2.4. **Computable Stone spaces.** Recall that a (countable, discrete) algebraic structure is computable if its domain, operations, and relations are uniformly computable. The result below was established in [23, 25]; see also [12, 11] for a detailed explanation.

Theorem 2.9. *For a Stone space S, the following are equivalent:*

(1) S has a computable (Polish) presentation.
(2) S has a computably compact presentation.

(3) The dual Boolean algebra $\widehat{S}$ has a computable presentation.

An algebraic structure is computably categorical if it has a unique computable presentation, up to computable isomorphism. A compact space is computably categorical if any two computably compact homeomorphic copies of the space are computably (bi-)homeomorphic (for computably compact presentations, “bi-” can be omitted because of Lemma 2.7(2)).

The following theorem is a combination of recent results from [1] and the well-known characterisation of computably categorical Boolean algebras due to Goncharov and Dzgoev [18] and LaRoche [32].

Theorem 2.10. *For a Stone space S, the following are equivalent:*

(1) S is computably categorical.
(2) $\widehat{S}$ is computably categorical.
(3) S has only finitely many isolated points.

In particular, we have:

Corollary 2.11. *The Stone spaces having a unique computably compact presentation up to (bi-)computable homeomorphism are exactly the following:*

(1) the Cantor space 2^ω,
(2) $2^\omega \sqcup F$, where F is finite, and
(3) finite spaces.

3. Degrees of 2^ω

3.1. **No least degree.** Corollary 2.11 implies that, up to (bi-)computable homeomorphism, 2^ω has exactly one computably compact presentation, *which we identify with the natural presentation of 2^ω by strings.* In other words, the *ct*-degree of (the natural presentation of) 2^ω is the only computably compact degree of the Cantor space.

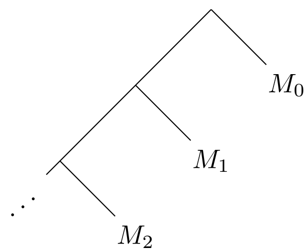
Since every computably compact presentation is minimal under $\leqslant_{ct}$ (Lemma 2.8), it is natural to ask whether the computably compact presentation of 2^ω is indeed the *least* under $\leqslant_{tc}$. The answer to this question is negative.

Theorem 3.1. *There exists a computable Polish homeomorphic copy M of 2^ω so that $M \not\geqslant_{ct} 2^\omega$.*

Remark 3.2. Note M cannot be *ct*-below 2^ω since it would make it computably compact by Lemma 2.7 and, thus, would imply $M \cong_{comp} 2^\omega$ by Corollary 2.11. Thus, $M \mid_{tc} 2^\omega$. We also remark that Theorem 3.1 (and its proof) will be used to prove the more general Theorem 1.5.

Proof. We begin with an informal description of M, and then we explain the basic strategy.

The description of M. The space will be a ‘fishbone’ consisting of clopen subcomponents M_i ($i \in \omega$), each M_i homeomorphic to 2^ω. More formally, we will have a uniformly computable sequence $(M_i)_{i\in\omega}$ of presentations of 2^ω, and M will be the 1-point compactification for the disjoint sum $\sqcup_i M_i$ of these presentations. The space will be metrised as follows. The distance between $x \in M_i$ and $y \in M_j$ will be set equal to $2^{-\max\{i,j\}}$ when $i \neq j$. When $i = j$, i.e., $x, y \in M_i$, the metric will be defined equal to the metric in M_i, but scaled down by 2^{-i-1}. Each M_i will have its diameter 1, but it will not be computably compact in general. The space can be visualised as 'tree' consisting of the infinite path 0^ω and also with a (scaled) M_i 'glued' to 0^i1, and having no other points.



Each M_i will be built as a c.e. closed subspace of the standard copy of the unit interval $[0,1]$. That is, during the construction, we will be uniformly listing computable points in $[0,1]$, and we will define M_i to be the closure of the resulting sequence. The flow of points in M_i will be controlled by the i^{th} diagonalisation strategy. Further, we will see that (depending on the outcome of the strategy) the space will be either a finite disjoint sum of copies of the Cantor space, or a 1-point compactification of ω copies of the Cantor space. In both cases, of course, $M_i \cong 2^\omega$. In the second case, however, the point that can be viewed as the limit point of 2^ω won't be computable uniformly in i. This feature will be used to diagonalise against the i^{th} potential reduction from M to the unique computably compact presentation of the Cantor space.

The basic strategy. We need to meet the requirements:

$$R_i : \ f_i \text{ does not witness } M \geqslant_{ct} 2^\omega,$$

Convention 3.3. We will monitor whether f maps fast Cauchy names to fast Cauchy names. If we discover f does not do this, by uniformly modifying the enumeration of all functionals, we can declare f to be diverging.

One iteration of the basic strategy is as follows. (We suppress i in M_i and f_i.)

(1) Begin building M with listing two (shrunk) copies of the Cantor set in $[0,1/3]$ and $[2/3,1]$, and assume $x_0 = 1/3$ and $y_0 = 2/3$ are also listed in these copies.
(2) Search for a basic clopen set $V \subseteq 2^\omega$ so that $f(x_0) \in V$ and $f(y_0) \notin V$.
(3) If such a V and precise enough approximations of $f(x_0)$ and $f(y_0)$ are ever found, declare x_0 'in' and y_0 'out'.
(4) Initiate the enumeration of a new (scaled down) copy of the Cantor set within an interval I_0 inside the sub-interval $[z_0, 2/3)$. Make sure the left-most endpoint z_0 of the sub-interval belong to this new copy.
(5) Wait for $f(z_0)$ to be placed either in or outside of V. If $f(z_0) \in V$ declare z_0 'in', and otherwise 'out'.
(6) Since x_0 is 'in' and y_0 is 'out', one of the two intervals
$$[x_0, z_0], [z_0, y_0]$$
will have its left endpoint 'in' and its right endpoint 'out'; say this interval is $[u, v]$.

(7) Declare $x_1 = u$ and $y_1 = v$.

Now iterate the basic strategy for x_1 and y_1, but using the same V that was already defined at the first iteration (that is, begin with (4) adjusting the notation mutatis mutandis).

In more detail, suppose that we have just defined x_n and y_n (so we have just seen z_{n-1} enter V or V^c). Define $z_n = \frac{x_n - y_n}{2}$, and enumerate z_n into M if it is not yet an element of $M[s]$. Because $M[s]$ is finite, there is a rational ε such that $[z_n, z_n - \varepsilon) \cap M[s] = \varnothing$, and such ε may be found computably. Take I_n of length $< \min\{\varepsilon/2, 2^{-n-1}\}$ with left endpoint z_n and initiate the enumeration of a Cantor set in I_n with z_n in the Cantor set. If there are points on both sides of z_n that were added for the same Cantor set, begin the construction of small Cantor sets around them in small intervals bounded away from all other enumerations of Cantor sets.

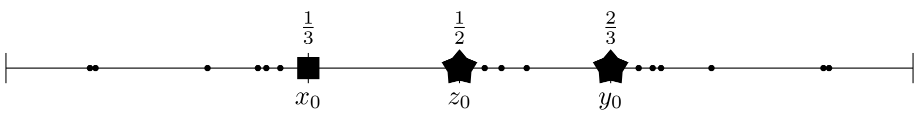


FIGURE 1. M_i. In this diagram, the squares are those whose images we have seen enter V; while the stars are those whose images we have seen are not in V. (The smaller points Cantor sets and hence ensure that M has the correct isomorphism type. It does not matter whether or not their images are in V.) Thus, in this example, $x_1 = x_0$ and $y_1 = z_0$ and the construction continues in $[x_0, z_0]$.

The *outcomes* of the basic strategy are as follows.

(1) Case 1: The basic strategy gets stuck after finitely many iterations.
(2) Case 2: The strategy is iterated infinitely many times.

We claim that in Case 1 the requirement is met. Indeed, the iterations in the basic strategy can get stuck for two reasons: (a) $f = f_i$ does not represent a computable map, or (b) V cannot be defined since $f(x_0) = f(y_0)$. In both cases f cannot be a computable homeomorphism.

We now argue that in Case 2 the requirement is met as well. Let ξ be the limit of the sequences $(x_i)_{i\in\omega}$ and $(y_i)_{i\in\omega}$ – note they converge to the same point. Observe that, for every i, $f(x_i) \in V$ and $f(y_i) \notin V$. Since V is clopen and f is continuous, we have that

$$V \ni \lim f(x_i) = f(\lim x_i) = f(\xi) = f(\lim y_i) = \lim f(y_i) \notin V,$$

which is a contradiction.

Remark 3.4. It may seem that Case 2 is impossible because of Convention 3.3. However, the situation is perhaps best understood using the definition of computability via effective continuity. Indeed, at any finite iteration, f can define $f^{-1}(V)$ and $f^{-1}(2^\omega \setminus V)$ to be two disjoint c.e. open sets, and it can delay their further enumeration. At the next iteration, it can carefully extend both of these c.e. sets while keeping them non-intersecting. This is where the lack of computable compactness is exploited in the strategy: we can always keep placing points outside of these c.e. open sets.

It appears that the infinite outcome is *unavoidable*: if we *could* somehow argue that the infinitary outcome (Case 2) is impossible, a mild modification of this strategy would allow us to construct a copy of $\mathrm{Intalg}(\omega)\widehat{}$ not above any computably compact copy, and this would contradict Theorem 1.5.

Construction. We begin with M being the 1-point compactification of $\bigsqcup_i M_i$ (as described earlier). Let the i^{th} strategy build M_i according to its instructions (as described in the basic strategy above).

Verification. There is no interaction or conflict between different strategies, and we have already argued that every requirement is isolation is met. It remains to argue that $M \cong 2^\omega$. For that, it is sufficient to demonstrate that $M_i \cong 2^\omega$, for every i. In case 1 of the i^{th} strategy we clearly have $M_i \cong \bigsqcup_{j \leqslant n} 2^\omega$, where n is the number of iterations of the strategy before it gets stuck. Otherwise, we shall define an infinite shrinking sequence of Cantor-line subsets in $[0,1]$ converging to ξ. In other words, in Case 2 the clopen subset M_i of M will be homeomorphic to the 1-point compactification of $\sqcup_{k \in \omega} 2^\omega$, where ξ plays the role of ∞. (If the reader is not closely familiar with Alexandroff's 1-point compactification, they may think of a 'fishbone' composed of infinitely many copies of 2^ω with 0^ω being ∞.) It is, of course, homeomorphic to 2^ω. □

Remark 3.5. We could build M witnessing Theorem 3.1 to be c.e. closed in $[0,1]$. For that, instead of forming a 'fishbone' out of the components M_i, realise them inside shrinking disjoint sub-intervals of $[0,1]$ converging to the point 0. Stone spaces realised as c.e. closed subsets of $[0,1]$ will play an important role later (Lemma 4.2).

3.2. **Embedding the c.e. Turing degrees into $\mathbf{CT}(2^\omega)$.** In this section we show that there is an embedding of the c.e. Turing degrees, $\mathcal{R}$, into $\mathbf{CT}(2^\omega)$. As a consequence of the proof, we show in Corollary 3.10 that the standard copy of Cantor space *branches*. Unfortunately, as we show in Theorem 3.9, this embedding does not preserve suprema, and we suspect that it fails to preserve infima as well. Nonetheless, we hope that by picking out a subclass of $\mathbf{CT}(2^\omega)$ for which our computability techniques are well-suited, that we can better understand the global properties of this degree structure.

Definition 3.6. For a c.e. set A, identify A with its characteristic function, and further identify the function with an infinite binary string. Under these conventions, $\alpha = 0.A$ is a left-c.e. real. Define

$$C(A) = \{x \in 2^\omega : x \leqslant_{\text{lex}} \alpha\}.$$

Observation 3.7. If A is non-computable, then α is clearly irrational, and thus $C(A) \cong 2^\omega$. Further, since α is left-c.e., we can list all finite strings σ so that $\sigma 0^\omega \in C(A)$; this shows that $C(A)$ is a computable metric space under the usual ultrametic on 2^ω. Finally, it is not difficult to see that, for a non-computable A, the resulting presentation $C(A)$ is *not* computably compact. (Indeed, if it were computably compact, it would give a way of computing α and, thus, A.)

$C(\cdot)$ gives the desired embedding of $\mathcal{R}$ into $\mathbf{CT}(2^\omega)$.

Theorem 3.8. *For c.e. sets A and B, $A \leqslant_T B$ if and only if $C(A) \leqslant_{ct} C(B)$.*

Notice that $C(\omega) = 2^\omega$, so $C(A) \geqslant_{ct} 2^\omega$ for every c.e. A.

Proof of Theorem 3.8. Suppose that $C(A) \leqslant_{ct} C(B)$ via f. Without loss of generality, A is not computable. Let $\xi = f^{-1}(\alpha)$, so $\alpha \leqslant_T \xi$. Towards a contradiction, suppose that $\xi \neq \beta$. Then by continuity, there is a clopen set $[\tau] \subseteq C(B)$ with $\beta \notin \tau$ and $\xi \in \tau$. Hence $[\tau]$ is bi-computably homeomorphic to 2^ω, so its image under the computable homeomorphism f is computably compact. But this implies A is computable, a contradiction.

The other direction is a generalisation of Example 1.4. We view $C(A)$ as a c.e. closed subset of 2^ω. For a string σ, let $[\sigma]$ be the clopen set of all extensions of σ in the space. We also identify σ with the infinite string $\sigma 0^\omega$. We assume throughout that A and B are infinite.

Let $A \leqslant_T B$ via Φ_e. Write α_s and β_s for the righmost paths in $C(A)[s]$ and $C(B)[s]$ respectively. Without loss of generality, we speed up the enumeration of $C(A)$ and $C(B)$ so that for every s, $\Phi_{e,s}^{\beta_s} \restriction |\alpha_s| = \alpha_s$, and so that there is a new "rightmost splitting" on $C(A)$ and $C(B)$ at every stage: i.e., there is $\sigma \prec \alpha_{s+1}$ with $\sigma 0, \sigma 1 \in C(A)[s+1] \backslash C(A)[s]$, and analogously for $C(B)$.

We shall describe one iteration of the procedure that builds a computable bijection $f : C(B) \to C(A)$. Recall that a continuous f is computable iff "f^{-1}(basic open) = c.e. open", and this is uniform. We shall exploit that the full pre-image of a clopen set in $C(A)$ does not have to be listed instantaneously and that this set-up guarantees that if α_s moves to the right, so must β_s.

Indeed, the rightmost path on each $C(A)$ and $C(B)$ are the only points that present any challenge. If $\sigma \in C(A)[s]$ and $\tau \in C(B)[s]$ are not initial segments of A_s and B_s respectively, then $[\sigma]$ and $[\tau]$ are computably bi-homeomorphic, uniformly, and we may choose f to copy this homeomorphism.

One iteration of the linking procedure. Initially, our convention implies that $C(A)[1]$ consists of a single split: $\sigma 0$ and $\sigma 1 = \alpha_s$, while $C(B)[s]$ may contain many leaves. We *link* $[\alpha_s]$ and $[\beta_s]$. This in effect declares that

$$f^{-1}([\alpha_s]) \supseteq [\beta_s].$$

Similarly, writing $(\tau_i)_{i<n}$ for the leaves of $C(B)[s]$ which are not β_s, we *permanently link* $[\sigma_0]$ and $\bigcup_{i<n}[\tau_i]$. This declares that

$$f^{-1}([\sigma_0]) = \bigcup_{i<n}[\tau_i].$$

This is the end of the *first phase* of the procedure.

At a later stage s, we may see that α_s or β_s has "moved to the right" due to an element entering A or B; i.e., $\beta_s >_{\text{lex}} \beta_1$. In this case, the linking procedure *requires performing its second phase.* In the construction, this may happen much later than the first phase, with other linking procedures acting at intermediate stages.

The second phase then proceeds as follows. Notice that since $\Phi_e^{\beta_1} \restriction |\alpha_1| = \alpha_1$, we must be in one of the following two cases.

Case 1. α_s has moved to the right and β_s has moved to the right
Case 2. β_s has moved to the right and $\alpha_1 \leqslant \alpha_s$.

In the first case, a simple argument using the continuity of the Turing functional Φ_e shows that if $\tau \leqslant \beta_s$ has been linked to σ in $C(A)$, then $\sigma \leqslant \alpha_s$. Thus, for $t < s$

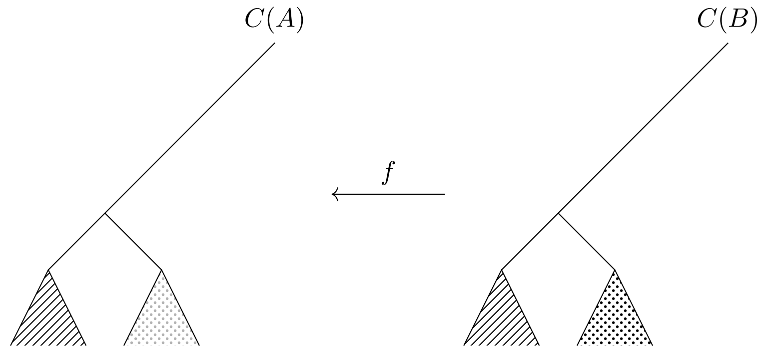


FIGURE 2. This is the set-up after Phase 1. f *permanently links* the interior nodes of $C(A)$ and $C(B)$. f *links* (denoted by grey colouring), but not permanently, nodes along the right-most path.

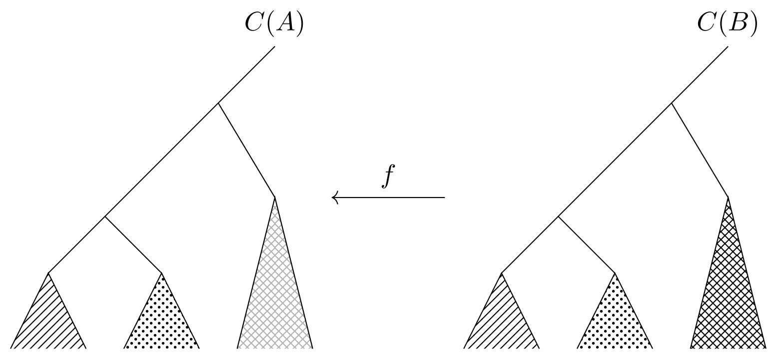


FIGURE 3. In Phase 2, after computations have settled, we may see (Case 1) new rightmost paths in $C(A)$ and $C(B)$. f then *permanently links* the interior nodes, including the nodes that were formerly along the rightmost path, and merely *links* (grey colouring) the new nodes along the rightmost path.

such that $\beta_t \prec \beta_s$ is maximal, it follows that $\alpha_t \prec \alpha_s$. (The converse may not hold in general.)

Thus, we may consistly link $[\alpha_s]$ and $[\beta_s]$, promising that

$$f^{-1}([\alpha_s]) \supseteq [\beta_s].$$

In the second case, $\alpha_1 \prec \alpha_s$, but β_s has "shifted to the right". Then link $[\alpha_s]$ to $[\beta_s]$. Furthermore, taking σ with $\alpha_1 \preccurlyeq \sigma \preccurlyeq \alpha_s$ witnessing a new "rightmost split" as guaranteed in our enumeration, we permanently link $[\sigma 00]$ to $[\beta_1]$. (We verify below that these links preserve other strategies.)

A few remarks about the procedure. The idea is, of course, to iterate the linking procedure to build a map $f : C(B) \to C(A)$. Before we proceed, we shall make two observations:

(O1) Note that the "internal" clopen sets in $C(A)$ mentioned in the linking procedure – i.e., those which do not include $[\alpha_s]$ – are uniformly computably

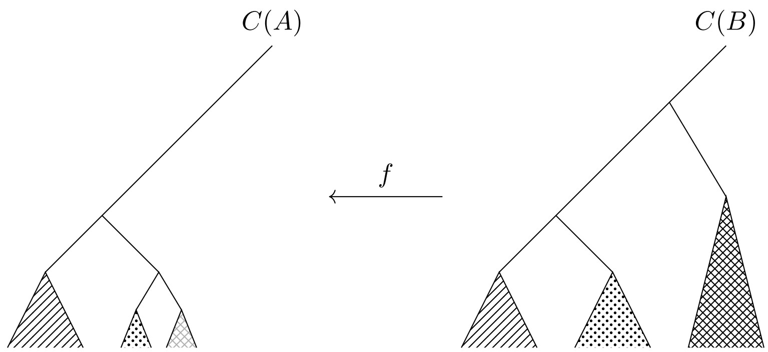


FIGURE 4. In Case 2 of Phase 2, we see a new rightmost path in $C(B)$, but not in $C(A)$. By convention, we do however see a new splitting in $C(A)$. This allows us to again *permanantly link* the interior nodes while creating room for the new open set in $C(B)$ to map into.

homeomorphic to 2^ω. This in particular includes the clopen set $\bigcup_{i<n}[\tau_i]$, which is not necessarily basic; however, it is a finite union of basic clopen sets, with all possible effective uniformity. It is clear how to (bi-)computably map it to 2^ω: simply represent 2^ω as a union of n clopen sets, and match those with the basic clopen sets $[\tau_i]$. Thus, once $[\tau] \subseteq C(B)$ is (not necessarily permanently) linked to $[\sigma] \subseteq C(A)$, where both $[\sigma]$ and $[\tau]$ are "internal", up to uniformly (bi-)computable homeomorphism, we can identify each with a copy of 2^ω.

(O2) Further, from the perspective of the linking procedure, $[\alpha\restriction s] \subseteq C(A)$ is essentially indistinguishable from $C(A)$: the only difference is that we replace A with A shifted by s. Naturally, this is true of $[\beta\restriction s] \subseteq C(B)$ as well. This is also uniform.

Increasing precision and making f surjective. Clearly, to ensure that our construction builds a computable bijective map, we must also explain how, given n and a better approximation of $x \in C(B)$, we output a 2^{-n}-approximation of $f(x)$. For this purpose, we need to declare relations of the form

$$f([\tau]) \subseteq [\sigma],$$

where the diameters of such τ decrease as the stage increases.

These conditions will collectively ensure that f is well-defined, computable, and bijective. However, simply recursively iterating the linking procedure will guarantee that these conditions are met. Indeed, every time we call the procedure, away from the rightmost paths, we link smaller clopen sets in $C(A)$ to smaller clopen sets in $C(B)$. Along the rightmost paths, we may occasionally have to link clopen sets in $C(A)$ to larger sets in $C(B)$, but this eventually settles since B is c.e.

We start by laying out the linking procedure in more detail for arbitrary clopen sets $U \subseteq C(A)$ and $V \subseteq C(B)$.

The linking procedure. As described above, the linking procedure $\mathrm{Link}(U, V)$ consists of 2 phases, where phase 2 is invoked when new elements are added to A or B below previously enumerated elements.

Phase 1: Increasing precision. Partition each of U and V into a two disjoint nonempty clopen sets "left" and "right", so $U = L_U \sqcup R_U$ and $V = L_V \sqcup R_V$. (If $U = [\sigma]$ and $V = [\tau]$, we will naturally choose $[\sigma 0]$ and $[\sigma 1]$ for L_U and R_U respectively, and analagously for L_V and R_V.) If neither U nor V contains α_s or β_s respectively, Phase 1 then permanently links L_U to L_V, and R_U to R_V.

If $\alpha_s \in U$ and $\beta_s \in V$, then $R_U = [\alpha_s]$ and $R_V = [\beta_s]$. In this case, Phase 1 permanently links $L_U = U \backslash [\alpha_s]$ and $L_V = V \backslash [\beta_s]$, while just linking $[\alpha_s]$ and $[\beta_s]$.

It is important that we only perform the linking procedure on clopen sets U and V that either are both "internal", as in the first case of Phase 1, or on the rightmost path, as in the second case.

Phase 2: Extending the domain. Phase 2 of $\mathrm{Link}(U, V)$ will be called at some stage s for which $\alpha_s, \alpha_{s+1} \in U$, $\beta_s, \beta_{s+1} \in V$ and $\beta_s \not\preccurlyeq \beta_{s+1}$. Furthermore, we assume that there are no $U_0 \subset U$ and $V_0 \subset V$ satisfying these conditions.

We have two cases, depending on whether $\alpha_s \preccurlyeq \alpha_{s+1}$ or $\alpha_s \not\preccurlyeq \alpha_{s+1}$.

In the first case, recall that by convention, there is $\alpha_s \preccurlyeq \sigma \prec \alpha_{s+1}$ with $\sigma 0$, $\sigma 1 \in C(A)[s+1] \backslash C(A)[s]$. Permanently link $[\sigma 00]$ to $[\beta_s]$ and link $[\alpha_{s+1}]$ to $[\beta_{s+1}]$.

In the second case, permanently link $[\alpha_s]$ to $[\beta_s]$ and link $[\alpha_{s+1}]$ to $[\beta_{s+1}]$.

Formal construction. The construction of f proceeds as follows.

Stage 0: Execute Phase 1 of $\mathrm{Link}(C(A), C(B))$, delaying Phase 2 until the next stage (if required by the strategy).

Stage $\mathbf{s+1}$: The beginning of the stage depends on we have seen new "small" elements enter B.

If $\beta_s \preccurlyeq \beta_{s+1}$**:** Perform Phase 1 of $\mathrm{Link}([\alpha_s], [\beta_s])$.

If $\beta_s \not\preccurlyeq \beta_{s+1}$**:** Let t be maximal with $\beta_t \preccurlyeq \beta_s$. Perform Phase 2 of $\mathrm{Link}([\alpha_t], [\beta_t])$.

Then recursively call Phase 1 of $\mathrm{Link}(U, V)$ for all pairs $U \subseteq C(A)[s+1]$, $V \subseteq C(B)[s+1]$ satisfying:

(1) $U \subseteq U_0$ and $V \subseteq V_0$, where U_0 is linked to V_0;
(2) U and V are finite unions of clopen sets with diameter $< 2^{-s}$;
(3) U and V are currently unlinked; and
(4) No subset of U or V is linked.

Continue this process until both $C(A)[s+1]$ and $C(B)[s+1]$ are completely covered by basic clopen sets of diameter $< 2^{-s}$, each contained in a linked clopen set.

Proceed to stage $s+2$.

Verification. We show that our construction builds $f : C(B) \to C(A)$ with the following properties.

1. *Consistency:* If U_0 is linked to V_0 and $V \supseteq V_0$ is linked to U, then $U \supseteq U_0$.
2. *Extendibility:* For every stage s, if U is linked to V and $V_0 \subseteq V$ is basic clopen with $V \backslash V_0 \neq \varnothing$ and V_0 unlinked, then there is $U_0 \subseteq U$ with U_0 not permanently linked to a clopen subset of $V \backslash V_0$. Additionally, if $U_0 \subseteq U$ is basic clopen with $U \backslash U_0 \neq \varnothing$ and U_0 unlinked, then there is $V_0 \subseteq V$ with V_0 not linked to a subset of $U \backslash U_0$.

(3) *Monotonicity:* If U is linked to V and U_0 is linked to $V_0 \subset V$, then $U_0 \subset U$.
(4) *Preservation of disjointedness:* If $V_0, V_1 \subseteq C(B)$ with $V_0 \cap V_1 = \varnothing$, then there are $U_0, U_1 \subseteq C(A)$ with U_i linked to V_i and $U_0 \cap U_1 = \varnothing$.
(5) *Precision refinement:* For every $n \in \omega$ and $\xi \in C(B)$, there is $\tau \preccurlyeq C(B)$ with $[\tau]$ linked to $U \subseteq C(A)$ with $\mathrm{diam}(U) \leqslant 2^{-n}$.
(6) *Covering:* For every n, there exists a stage t and finite collection of disjoint clopen sets $\{U_i\}_{i=0}^k \subseteq C(A)$ such that:
   (a) $\mathrm{diam}(U_i) < 2^{-n}$ for each i;
   (b) $\bigcup_i U_i = C(A)$;
   (c) Each U_i is linked to some $V_i \subseteq C(B)$.

These properties suffice for showing that f is a computable bijection: Given $\xi \in C(B)$, we compute $f(\xi)$ with precision 2^{-n} using property 5. Crucially, this computation is unambiguous since linking relations, once established, are permanent. By property 3, taking the limit of this process yields an effective computation of $f(\xi)$ from ξ. Finally, property 4 ensures injectivity of f, while property 6 guarantees surjectivity.

Thus it remains to verify that (1)–(6) hold. Each of the conditions are readily verified once U and V do not contain the rightmost path of $C(A)[s]$ and $C(B)[s]$. Thus, in what follows, we focus on the case when U and V contain α_s and β_s respectively.

In the construction, Phase 1 of $\mathrm{Link}(U, V)$ is only applied to U and V with no linked subsets, and hence preserves 1 and 3. Moreover, the requirement that $L_U, L_V, R_U, R_V \neq \varnothing$ ensures that Phase 1 preserves 2. 4 is guaranteed by our choice of $L_U \cap R_U = \varnothing$ and $L_V \cap R_V = \varnothing$.

Phase 2 is called to $\mathrm{Link}(U, V)$ when elements entering B cause β_{s+1} to shift to the right, and $V = [\beta_t]$, $U = [\alpha_t]$, where t is maximal with $\beta_t \preccurlyeq \beta_{s+1}$. Since B is c.e., β_s also extends β_t, so, by continuity of the Turing functional, $\alpha_s, \alpha_{s+1} \in U$. This guarantees that Phase 2's action of linking $[\alpha_{s+1}]$ to $[\beta_{s+1}]$ and $U \backslash [\alpha_{s+1}]$ to $V \backslash [\beta_{s+1}]$ preserves 3.

1 and 2 follow from the convention that there are new basic clopen sets $[\sigma 0] \in C(A)[s+1]$ and $[\tau 0] \in C(B)[s+1]$ where $\sigma \preccurlyeq \alpha_{s+1}$ and $\tau \preccurlyeq \beta_{s+1}$. Thus, neither U nor V is covered by its linked subsets at the end of Phase 2.

4 follows by the choice of linking pair for $[\beta_s]$, depending on whether $\alpha_s \preccurlyeq \alpha_{s+1}$.

Finally, since at the end of stage s, $C(A)[s]$ and $C(B)[s]$ are covered by basic clopen balls of diameter $\leqslant 2^{-s}$ which are contained in linked sets, 5 holds. 6 follows by waiting until α_s has stabilised.

This concludes the proof of Theorem 3.8. □

We finish this section with a couple of extended comments about the embedding $C(\cdot)$. We show at the end of the section (Corollary 3.10) that it implies the existence of a minimal pair over 2^ω. Of course, one would hope that $C(\cdot)$ preserves infima, and thus that the existence of a minimal pair in $\mathcal{R}$ would immediately imply one in $\mathbf{CT}(2^\omega)$. However, we expect that $C(\cdot)$ does not preserve infima; our first observation, in Theorem 3.9, is that for any $A|_T B$ in $\mathcal{R}$, $C(A)$ and $C(B)$ do not have a supremum in $\mathbf{CT}(2^\omega)$.

Theorem 3.9. *In $\boldsymbol{CT}(2^\omega)$, whenever $A \mid_T B$, the spaces $C(A)$ and $C(B)$ have no least upper bound.*

Proof. Let $C_1 = C(A \oplus B)$ and $C_2 = C(A) \sqcup C(B)$. Both are upper bounds. So there must be some M below both, say

$$g_1 : C_1 \to M, \quad g_2 : C_2 \to M, \quad f_1 : M \to C(A), \quad f_2 : M \to C(B).$$

Write α and β for the right-most paths of $C(A)$ and $C(B)$, respectively, and let $\tilde{\alpha}$ and $\tilde{\beta}$ denote their copies in C_2. Write γ for the right-most path in $C(A \oplus B)$; slightly abusing notation, we could also write $\alpha \oplus \beta$.

Let $x = f_1^{-1}(\alpha)$ and $y = f_2^{-1}(\beta)$, both elements of M. We are going to argue that neither $x = y$ nor $x \neq y$ can possibly hold, which would imply that such an M does not exist.

Case 1. $x \neq y$.

We have already observed earlier that the computable homeomorphic image of γ in $C(A)$ has to be α, so in particular

$$f_1 g_1(\gamma) = \alpha,$$

and thus, by the injectivity of all these maps,

$$g_1(\gamma) = x.$$

But essentially the same argument, this time applied to $C(B)$, shows that

$$f_2 g_1(\gamma) = \beta,$$

and therefore $g_1(\gamma) = y$ as well. Thus, $x = y$, and Case 1 is impossible.

Case 2. $x = y$.

Note that x computes both α and β, and thus can compute $A \oplus B$ as well. (More formally, its name $N^x = \{B : x \in B\}$ can enumerate the complement of the c.e. set $A \oplus B$, but this is the same as what we said for all intents and purposes.)

Now, the $(f_1 \circ g_2)$-preimage of α in C_2 has to be $\tilde{\alpha}$ or $\tilde{\beta}$, and in fact it has to be $\tilde{\alpha}$. But $f_1^{-1}(\alpha) = x$, so it follows that

$$g_2(\tilde{\alpha}) = x,$$

making x computable relative to A.

But we have just seen that x can compute $A \oplus B$.

So Case 2 is also impossible. □

We now pay off our final debt of this section and proceed with a slightly more involved argument to show that 2^ω branches.

Corollary 3.10. *There exist* $\mathbf{a}, \mathbf{b} \in \boldsymbol{CT}(2^\omega)$ *so that* $\mathbf{a}|\mathbf{b}$ *and* $\inf\{\mathbf{a}, \mathbf{b}\} = \mathbf{deg}\ (2^\omega)$.

The proof of Corollary 3.10 relies on enumeration reducibility.

Recall that $A \leqslant_e B$ informally means A can be enumerated using only positive information about B. More precisely:

Definition 3.11 (Enumeration Reducibility)**.** Let A and B be sets of natural numbers. We say that A is *enumeration reducible* to B (denoted $A \leqslant_e B$) if there exists a computably enumerable (c.e.) set Φ of pairs $\langle n, D \rangle$, where $n \in \mathbb{N}$ and D is a finite subset of $\mathbb{N}$, such that:

$$n \in A \leftrightarrow \exists D \subseteq B \ \ \langle n, D \rangle \in \Phi.$$

Here, Φ is called an *enumeration operator*, and $A = \Phi(B)$.

Another way to interpret $A \leqslant_e B$ via Φ is that Φ turns any (not necessarily effective) enumeration of B into some enumeration of A.

The least enumeration degree, $\mathbf{0_e}$, is the degree consisting of all c.e. sets. A degree is Π^0_1 if it contains the complement of a c.e. set.

The enumeration degrees are fairly well-studied; we will need only one further well-known result that can be found in [51] (Theorem 4.1 on p 317 therein). It says that there exists a minimal pair of Π^0_1 enumeration degrees:

Theorem 3.12. *There exist Π^0_1 sets A and B such that:*

(1) $A \,|_e\, B$ (i.e., A and B are e-incomparable), and
(2) $\mathbf{0_e} = \inf\{deg_e(A), deg_e(B)\}$.

We shall also need an elementary lemma that relates enumeration degrees with computable topological reducibility. Recall that $FinCov(M)$ consists of all finite covers of M by basic open sets.

Lemma 3.13. *Suppose $A \leqslant_{ct} B$. Then $FinCov(A) \leqslant_e FinCov(B)$.*

Proof. Let $f : B \to A$ be the computable bijection witnessing $A \leqslant_{ct} B$. Let $\vec{U}$ be a finite collection of basic open sets in A. (As usual, we abuse notation and identify basic open sets with their indices.) Evidently $\vec{U} \in FinCov(A)$ iff $f^{-1}(\vec{U})$ covers B.

Recall that $f^{-1}(U_i)$ is c.e. open for each $U_i \in \vec{U}$; thus we write $f^{-1}(\vec{U})[s]$ to denote the subset of $f^{-1}(\vec{U})$ enumerated by stage s. Since B is compact, $f^{-1}(\vec{U})$ covers B iff there is s such that $f^{-1}(\vec{U})[s]$ covers B.

Since $f^{-1}(U)$ is c.e. open uniformly in U, $FinCov(A) \leqslant_e FinCov(B)$. □

Remark 3.14. Since M from Theorem 3.1 is not computably compact, $FinCov(2^\omega) = \mathbf{0_e} <_e FinCov(M)$, yet $M \,|_{tc}\, 2^\omega$ (Remark 3.2). Thus Lemma 3.13 cannot be strengthened to an "iff" characterisation.

Proof of Corollary 3.10. We begin by showing that

$$(\dagger) \qquad FinCov(C(A)) \equiv_e \overline{A}.$$

To determine if $x \in \overline{A}$, wait for a finite cover of $C(A)$ witnessing that the rightmost path at level x does not contain 1. On the other hand, any enumeration of $\overline{A}$ can be used to compute A and hence $C(A)$. This can be turned to a uniform list of 2^{-i}-covers of $C(X)$, one for each i. By Proposition 2.3 (partially relativised), we can use this list to enumerate $FinCov(C(X))$.

Now let A and B be c.e. sets whose complements $\overline{A}$ and $\overline{B}$ form a minimal pair in the Π^0_1 e-degrees, as given by [51, Theorem 4.1]. By Theorem 3.8, $C(A)$, $C(B) \geqslant_{ct} 2^\omega$.

By Lemma 3.13 and (†), $C(A)|_{ct}C(B)$. Further, if $M \leqslant_{ct} A, B$ then, by the choice of the sets and the same lemma, $FinCov(M)$ has to be c.e.; that is, M has to be computably compact. By Corollary 2.11 M has to be bi-computably homeomorphic to 2^ω. By Lemma 2.7, $M \equiv_{tc} 2^\omega$. □

3.3. **The Schröder–Bernstein Theorem fails for $\leqslant_{ct}$.** Recall that $A \equiv_{ct} B$ means that there exists a computable homeomorphism from A to B, and a computable homeomorphism from B to A. One naturally wonders whether an analogy of Schröder–Bernstein Theorem can be established. That is, does $A \equiv_{ct} B$ imply that there is a homeomorphism $f : A \to B$ so that *both* f and f^{-1} are computable?

In the latter case we shall write $A \cong_{comp} B$ and say that A and B are *bi-computably homeomorphic*. Evidently, $\cong_{comp}$ implies $\equiv_{ct}$. However, the implication is strict as we now show.

Theorem 3.15. *There are two effective copies of the Cantor space which are both effectively reducible to each other, but there is no bi-computable homeomorphism.*

Proof. We will use the spaces $C(A)$ defined in Definition 3.6. We start by finding appropriate A to which to apply $C(\cdot)$.

Lemma 3.16. *There exists a uniformly c.e. sequence of sets $(A_i)_{i\in\omega}$ with the properties:*

(1) the sets form a $\mathbb{Z}$-chain under set-theoretic inclusion;
(2) the sets form a $\mathbb{Z}$-chain under $<_T$;
(3) $(A_i)_{i\in\omega}$ admits a uniformly synchronised enumeration.

Proof. We shall exploit the well-known Sacks Splitting and Density Theorems that can be found in [50]. Fix any non-computable, incomplete c.e. set A_0. To obtain an ω^*-chain with the desired properties, iterate the Sacks Splitting Theorem between A_0 and $\mathbf{0}$ (e.g., always choose the left half of the split). Note that the enumeration for the chain is also synchronised. Without loss of generality, we may assume that all these sets contain only even numbers.

Split the odd numbers into an infinite collection of non-intersecting computable sets $(U_i)_{i\in\omega}$. Then extend it to a $\mathbb{Z}$-chain as follows. To define A_1, use the Density Theorem to uniformly fix a c.e. set B_0 so that $A_0 <_T B_0 <_T K$, so that $B_0 \subseteq U_0$ and consider $A_1 = A_0 \sqcup B_0$. Iterate this to define $B_1 \subseteq U_1$ between A_1 and K, and $A_2 = A_1 \sqcup B_1$, and so on. It should be clear that the resulting sequence satisfies the desired properties. □

Fix (A_i) from the lemma above; note all of these sets are non-computable, and thus $C(A_i) \cong 2^\omega$, for each i. For $i \in \mathbb{Z}$, define

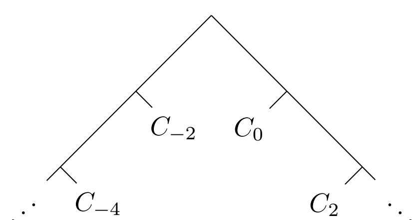
$$C_i = C(A_i),$$

where (A_i) is the sequence from the lemma above. The real

$$\alpha_i = 0.A_i,$$

is the "rightmost path" through the tree $T_i \subseteq \{0,1\}^{<\omega}$ so that $C_i = [T_i]$. Observe that, by Theorem 3.8, for $i > j$, there is an effective homeomorphism $C_i \to C_j$.

Let M be the space described by:

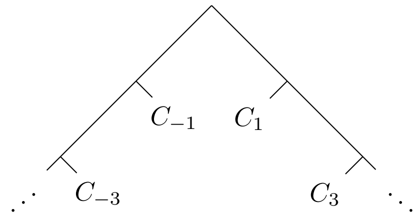


That is,

$$M = \{1^n 0^\frown x : x \in C_{2(n-1)}\} \cup \{0^n 1^\frown x : x \in C_{-2n}\} \cup \{0^\infty, 1^\infty\}.$$

As the C_i are uniformly computable, this is a computable space.

Similarly, let N be the space described by:



Then there is a computable homeomorphism $N \to M$ mapping each C_{2n+1} to C_{2n}. Similarly, there is a computable homeomorphism $M \to N$ mapping each C_{2n} to C_{2n-1}.

Let us abuse notation by writing α_i for the rightmost element of the copy of C_i in M or N (depending on the parity of i). (Note that technically the rightmost point of the copy of, for example, C_0 is actually $10^\frown\alpha_0$.)

Towards a contradiction, suppose there is a bi-computable homeomorphism $f : M \to N$. Since f is *bi*-computable, $f(x) \equiv_T x$ for all $x \in M$, so it cannot be that $f(\alpha_0) = \alpha_i$ for any (odd) i. Nor can $f(\alpha_0) = 0^\infty$ or 1^∞. Thus $f(\alpha_0)$ must be an "interior" point of some C_i (i.e., must be contained in some clopen set to the left of α_i).

By continuity of f, there is some $\sigma \not\prec \alpha_i$ and some $\tau \prec \alpha_0$ with $f([\tau]) \subseteq [\sigma]$. Note that $[\sigma] \cap N = [\sigma]$, and is thus it is computably compact. Since by assumption f^{-1} is also computable, $f^{-1}([\sigma])$ is computably compact, by Lemma 2.7(1). Thus, given n, we can list basic 2^{-n}-balls covering $f^{-1}([\sigma])$. Noticing that if $\eta \in C_0$ and $\rho \in M \backslash C_0$, then $d(\eta, \rho) \geqslant 2^{-3}$, we observe that basic clopen U of M with diameter $< 2^{-3}$ satisfies that either $U \subseteq C_0$ or $U \cap C_0 = \varnothing$, and which case U falls into can be determined uniformly.

Hence from the computable compactness of $f^{-1}([\sigma])$, we obtain a uniform enumeration of the finite covers by basic clopen sets of diameter 2^{-n} of $f^{-1}([\sigma]) \cap C_0$. Combining this with a listing of the 2^{-n}-covers of $C_0 \backslash [\tau]$ (which we can obtain without any assistance from f^{-1}), we obtain an enumeration of the finite 2^{-n}-covers of C_0, uniform in n.

But this is impossible, since α_0 is a non-computable real and, thus, C_0 is not computably compact. $\square$

4. The 1-atom

Let $\mathcal{A}_1$ be the unique (up to homeomorphism) compact space having exactly one limit point. It is the Stone space $\widehat{\mathrm{Intalg}(\omega)}$ of the Boolean algebra $\mathrm{Intalg}(\omega)$, generated by half-open, half-closed intervals of ω. Alternatively, it is the one-point compactification of the discrete countable space. It is also the Π^0_1 class consisting of 0^∞ and $0^n 1^\infty$ ($n \in \omega$) and commonly referred to as the "fishbone". To give $\mathcal{A}_1$ a better name, we shall mix the terminology of Boolean algebras with topology and refer to $\mathcal{A}_1$ as the "1-atom". (Of course, it would be more appropriate to refer to the dual Boolean algebra $\mathrm{Intalg}(\omega)$ as the 1-atom.)

The space $\mathcal{A}_1$ will play a central role in several characterisations, and the techniques associated with $\mathcal{A}_1$ are quite different from those that arise in the study of 2^ω.

4.1. **Making $\mathcal{A}_1$ computably compact.** By [5], $\mathcal{A}_1$ is not *computably categorical* – i.e., does not have a unique computably compact copy – and hence does not have a minimum degree. However, the main result of this section is that every element of $\mathbf{CT}(\mathcal{A}_1)$ is *ct*-above some computably compact copy. This is in contrast to the case of $\mathbf{CT}(2^\omega)$, as shown in Theorem 3.1.

Theorem 4.1. *Every copy of $\mathcal{A}_1$ is ct-above some computably compact copy.*

Proof. We begin with a lemma that contrasts with the earlier Remark 3.5.

Lemma 4.2. *Suppose C is a copy of the 1-atom realised as a c.e. subset of the reals (w.l.o.g. of [0,1], up to computable scaling). Then C is ct-above some computably compact copy.*

We first explain how to use the lemma to prove the theorem, and then we prove the lemma.

Let M be the given copy of $\mathcal{A}_1$. Non-uniformly scale the metric in M by an integer to make sure the diameter is < 1, and then computably homomorphically embed M into the Hilbert Cube $\mathcal{H} = [0,1]^\omega$ via

$$x \mapsto (2^{-i} d(x, x_i))_{i \in \omega},$$

where (x_i) is the fixed computable dense sequence in M. As mentioned in the preliminaries section, we may assume w.l.o.g. the sequence is without repetition.

Let D be the computable image of M in $\mathcal{H}$. We evidently have $M \geqslant_{ct} D$; thus, for the purposes of this proof we may identify M with D and injectively and effectively map $M \subseteq \mathcal{H}$ to $[0,1]$. If we succeed, the lemma will imply the theorem.

Consider the projection $\pi_1(M)$ of M onto the first 'coordinate' of $\mathcal{H}$, which is a copy of $[0,1]$. Having in mind Lemma 4.2, we'd wish this projection be injective on M. Of course, it is not injective in general. The next lemma partially fixes it on all points, except perhaps the limit point.

Lemma 4.3. *There exists $X \subseteq \mathcal{H}$ so that $X \leqslant_{ct} M$ and $\pi_1 \restriction_X$ is injective on the special points of X.*

We postpone the proof of Lemma 4.3 too, and proceed with the proof of Theorem 4.1 assuming the lemmata. As before, we may identify M with X and assume π_1 is injective on the special points of M.

If the unique limit point ξ of M is a special point or the map π_1 happens to be injective on all of M (including ξ), we are done by Lemma 4.2 and the elementary fact that any injective continuous map between compact spaces is a homeomorphism. If not, suppose the projection is not injective on the entirety of M. This is possible only when

$$\pi_1(\xi) = \pi_1(x_i),$$

for some (and only one such) special point x_i. But x_i is isolated from the rest by a *clopen* basic ball B whose rational radius r can be obtained non-uniformly. All other points of the space are in the formal complement $\{x : d(x, x_i) > r\}$ of $B = \{x : d(x, x_i) < r\}$. Now modify π_1 to get an injective embedding $f : M \to [0,1]$ as follows.

Pick any rational $y \in [0,1] \setminus \pi_1(M)$ – note it actually exists since $\pi_1(M)$ is countable closed. If z is in $M \setminus B$, then set $f(z) = \pi_1(z)$. Otherwise (in which case of course $z = x_i$) set $f(z) = y$.

We see that f is clearly a computable map (because $M\backslash B$ is c.e. open), and it is also evidently injective. We have that $f(M) \leqslant_{ct} M$ and $f(M) \subseteq [0,1]$, and the theorem follows from Lemma 4.2.

This finishes the proof of Theorem 4.1 assuming Lemmas 4.2 and 4.3. It remains to prove the lemmas.

Proof of Lemma 4.2. So suppose $C = cl(x_i)$, where (x_i) is a computable sequence of computable points in [0,1]. It is not too hard to see that, by scaling the metric in [0,1] by a computable real, we can assume that the distances between x_i and x_j are pairwise irrational; e.g., [22] and [11, Exercise 2.4.33]. Such scaling is, of course, a computable homeomorphism.

There are two cases:

Case 1. The unique limit point ξ of C is computable. Without loss of generality, we can just assume $\xi = x_0$ by adjoining it to the sequence, if necessary, from the very beginning.

Case 2. Not Case 1. In this case observe that the distance to any x_j is not computable, *since it would make* ξ *a computable point in* $[0,1]$. (The reader should take a moment to appreciate this observation.)

In both cases, but for different reasons, no point of the space can lie at the formal boundary of the basic open ball $B(x_i, r)$:

$$\{y : d(x_i, y) = r\},$$

where r is a positive rational. Indeed, we handled all special points using scaling, and Case 2 handles the case when ξ is not special/computable in C.

Now, we define nice clopen splits of C, as follows. Simply list pairs $(B(x_i, r), C - B(x_i, r))$, where $C - B(x_i, r)$ is given by the enumeration of all basic open balls formally disjoint from $B(x_i, r)$. Since no point of the space lies at the formal boundary of $B(x_i, r)$, both c.e. open sets are complements of each other. These clopen partitions separate arbitrary points of the space, and thus form a base.

Fix a uniform enumeration of all (by the above analysis, clopen) basic balls (B_i) in C, as well as their (formal) complements $(C - B_i)$; write B_i^0 for $C - B_i$ and B_i^1 for B_i. Without loss of generality, we may assume $B_0 = C$.

Note that the relation $\bigcap_{i\in I} B_i^{\varepsilon_i} = \varnothing$ is co-c.e. The rest of the proof now (essentially) follows the proof of a folklore result stating that every Π^0_1-presented Boolean algebra has a computable presentation (e.g., [47]); the difference is that we need to work with the duals and also control the isomorphism.

We will define a computable tree $T \subseteq 2^{<\omega}$ with no dead ends and set $N = [T]$ and a computable homeomorphism $f : C \to N$. The idea is to split the tree T "very very late" (far down below) in case if some clopen set is discovered to be non-empty.

First, define a c.e. binary tree Γ as follows. Associate each clopen $\bigcap_{i<n} B_i^{\varepsilon_i}$ to the finite string σ defined by $\sigma(i) = \varepsilon_i$ and let D_σ be another notation for $\bigcap_{i<n} B_i^{\varepsilon_i}$. Enumerate σ into Γ if $D_\sigma \neq \varnothing$. Notice that there is natural homeomorphism $g : C \to [\Gamma]$ sending x to σ where $x \in \bigcap_{i\in\omega} B_i^{\sigma(i)}$.

However, $[\Gamma]$ may not be computable. To define a computable T, we inductively "stretch" Γ and also push down the splits if they occur in Γ too late.

At stage s, suppose T_s has been defined. To ensure that T is computable, we will not add any new sequences of length s to T. Thus, it will be necessary to ensure at the end of the stage that every leaf has length at least $s+1$ so that T has no dead ends.

First "stretch" the tree by adjoining $\sigma\hat{}0$ for every leaf $\sigma \in T_s$.

To delay the splits, assume (say) $\tau\hat{}1$ is enumerated in Γ at stage s, for some τ that was already in Γ. Suppose that τ is *linked* with σ in T, as in the proof of Theorem 3.8; i.e., we have declared $f([\tau]) \subseteq [\sigma]$ for the homeomorphism $f : [\Gamma] \to [T]$ that we are defining. If $\tau\hat{}0$ is not in Γ_s, then associate $[\tau\hat{}1]$ with $[\rho]$, where ρ is the leaf of T_s extending σ. (The construction will ensure that there is a unique such ρ since τ does not branch in Γ_s.) If $\tau\hat{}0 \in \Gamma_s$, then, as in the proof of Theorem 3.8, we choose a fresh $\rho \succ \sigma$ and permanently link $\tau\hat{}1$ with $\rho\hat{}1$ (that is, declare $f([\tau\hat{}1]) \subseteq [\rho\hat{}1]$). We also link $\tau\hat{}0$ with $\rho\hat{}0$. Thus, we are declaring

$$f^{-1}([\rho\hat{}1]) = [\tau\hat{}1]$$

and

$$f^{-1}([\rho\hat{}0]) \supseteq [\tau\hat{}0]$$

In this way we proceed to define a computable homeomorphism f from $[\Gamma]$ to $[T]$, using an iterated construction closely resembling that in Theorem 3.8.

Note that $[\Gamma]$ and C share essentially the same notation for their bases of topology: D_σ is in a 1-1 correspondence with $\sigma \in \Gamma$, and this in turn gives rise to a 1-1 correspondence between D_σ and $[\sigma] \subseteq [\Gamma]$. It follows that f can be re-interpreted as a computable homeomorphism from C to $[T]$, as required. □

Proof of Lemma 4.3. We view $\mathcal{H}$ as a computable closed subset of a computable Hilbert space. (Since ℓ_2 is actually computably categorical with respect to linear isometry [38], it does not matter which copy we fix, as long as we use the standard norm. In particular, we may view $\mathcal{H}$ as a subset of ℓ_2 represented by sequences with finite support.) For instance, the usual linear structure on the space is also computable, the notation $2\mathcal{H}$ makes sense, and furthermore $2\mathcal{H}$ is bi-computably homeomorphic to $\mathcal{H}$ since scaling by 2 is evidently a computable homeomorphism.

Let (x_i) be an effective (in $\mathcal{H}$) sequence which is dense in $M \subseteq \mathcal{H}$. It is sufficient to construct a computable injective map $h : \mathcal{H} \to 2\mathcal{H}$ and meet the requirements

$$R_k : \pi_1 h(x_i) \neq \pi_1 h(x_j), \text{ where } i \neq j, \text{ and } i, j \leqslant k,$$

for every k.

Local shift functions. Let $I = [a, b] \subseteq [0, 1]$ be an interval such that $|a - b| < \varepsilon$. Define the function $s_{I,\varepsilon} \colon [0, 1] \to \mathbb{R}$ by:

$$s_{I,\varepsilon}(x) = \begin{cases} 0, & \text{if } x \leqslant a, \\ \varepsilon \cdot \dfrac{x-a}{b-a}, & \text{if } a < x < b, \\ \varepsilon, & \text{if } x \geqslant b. \end{cases}$$

For a point $x = (\pi_i(x))_{i \in \mathbb{N}} \in \mathcal{H}$ and $j > 1$, define

$$\hat{s}_{I,\varepsilon,j}(x) = (\pi_1(x) + s_{I,\varepsilon}(\pi_j(x)), \pi_2(x), \pi_3(x) \ldots, \pi_i(x), \ldots),$$

which results in "shifting" all points of $\mathcal{H}$ having their π_j-coordinates $\geqslant a$ by at most ε "to the right" along the π_1-coordinate. We intend to iterate such functions using various parameters; we write $\bigcirc_{i=0}^{k} f_k$ to denote the composition $f_k(f_{k-1}(\dots f_0(x)))$.

Note that if (ε_k) are so that $\varepsilon_k < 2^{-k}$ then any sequence of maps of the form

$$S_k(x) = \bigcirc_{i=0}^{k} \hat{s}_{I_i,\varepsilon_i,j_i}(x), \quad k \in \omega$$

is fast converging (viewed as a sequence in the computable Banach space $C(\mathcal{H}, \ell_2)$), and so its limit is a computable function; denote it f. We claim that any such f is furthermore injective. Indeed, we have that

$$f(x) = (\pi_1(x) + \sum_{k=0}^{\infty} s_{I_k,\varepsilon_k}(\pi_{j_k}(x)), \pi_2(x), \pi_3(x) \dots, \pi_i(x), \dots).$$

Suppose $x \neq y$. Then either $\pi_i(x) \neq \pi_i(y)$ for some $i > 1$, and then

$$\pi_i(f(x)) = \pi_i(x) \neq \pi_i(y) = \pi_i(f(y)),$$

or otherwise $\pi_i(x) = \pi_i(y)$ for all i. If so, it must be that $\pi_1(x) \neq \pi_1(y)$. But we also have that

$$\sum_{k}^{\infty} s_{I_k,\varepsilon_k}(\pi_{j_k}(x)) = \sum_{k}^{\infty} s_{I_k,\varepsilon_k}(\pi_{j_k}(y)),$$

and therefore $\pi_1(f(x)) \neq \pi_1(f(y))$ in this case too.

We see that, if we restrict ourselves to iterative applications of such shift-functions, we preserve injectivity in the limit, on the entire $\mathcal{H}$. By choosing ε_k small (i.e., $\sum_k \varepsilon_k < 1$), we can additionally guarantee that the limit of the process is in $2\mathcal{H}$.

The construction of h. We construct h using iterative applications of shift-functions, as was explained above. In the construction, we only specify the choices of the parameters ε_k, I_k, and j_k.

To meet R_1, we must make sure $\pi_1(x_0) \neq \pi_1(x_1)$; recall that we assume $x_i \neq x_j$ throughout. If we discover that $\pi_1(x_0) \neq \pi_1(x_1)$, we don't have to do anything. Otherwise, we must find a $j = j_1$ so that $\pi_j(x_0) \neq \pi_j(x_1)$. We can wait for as long as we desire, until we either see $\pi_1(x_0) \neq \pi_1(x_1)$ (in which case do nothing, but impose restraint on later actions, as described below), or we have that there is a j with $|\pi_j(x_0) - \pi_j(x_1)| > 2\varepsilon_1$ and $|\pi_1(x_0) - \pi_1(x_1)| < \varepsilon_1/2$, where ε_1 is as small as we desire.

Then we choose a small (in particular, smaller than ε_1) interval I_1 between $\pi_j(x_1)$ and $\pi_j(x_0)$, and define $S_1 = \hat{s}_{I_1,\varepsilon_1,j}(x)$ so that

$$\pi_1 S_1(x_1) = x_1 + \hat{s}_{I_1,\varepsilon_1,j}(x_1) \neq x_1 + \hat{s}_{I_1,\varepsilon_1,j}(x_0) = \pi_1 S_1(x_0),$$

and indeed the difference is at least $\varepsilon_1/2$. (We intend to keep it at least $\varepsilon_1/4$ by choosing all our future ε_i $(i > 1)$ very small that their collective sum is at most $\varepsilon_i/4$.) If we are in the case in which we have verified $\pi_1(x_0) \neq \pi_1(x_1)$, then we shall also keep the sum of all our future epsilons smaller than $|\pi_1(x_0) - \pi_1(x_1)|$. In both cases, R_1 will be met in the limit.

At the end of this stage, we shall define h_1 to be either S_1 or the identity, and we also restrict our future choice of ε_i, $i > 1$, as described in the previous paragraph, depending on whether we have discovered $\pi_1(x_0) \neq \pi_1(x_1)$ or found a parameter j_1 with $|\pi_{j_1}(x_0) - \varphi_{j_1}(x_1)| > 2\varepsilon_1$. Note h_1 is injective.

To meet R_2, R_3 (etc.), we iterate this process. For instance, to meet R_2, we apply a similar procedure to $h_1(\mathcal{H})$, $h_1(x_2)$ and $h_1(x_0)$ to satisfy

$$\pi_1(h(x_2)) \neq \pi_1(h(x_0))$$

(while maintaining $\pi_1(h_2(x_1)) \neq \pi_1(h_2(x_0))$ by keeping all parameters small). This way we shall define h_2. To satisfy $\pi_1(h(x_2)) \neq \pi_1(h(x_1))$ (while maintaining the previously met inequalities by controlling the epsilons), we will apply a similar procedure to $h_2 h_1(\mathcal{H})$, and so on.

As we have already explained earlier, the process converges to h which is a computable injective map homomorphically mapping $\mathcal{H}$ to $2\mathcal{H}$. The desired space X is $\frac{1}{2} \cdot h(\mathcal{H}) \subseteq \mathcal{H}$. □

□

4.2. **Common upper bound.** Recall that $\mathcal{A}_1$ denotes the unique (up to homeomorphism) compact space having exactly one limit point.

Theorem 4.4. *In* $\mathbf{CT}(\mathcal{A}_1)$*, every pair of degrees has a common upper bound.*

We leave open the existence of least upper bounds of pairs of degrees in $\mathbf{CT}(\mathcal{A}_1)$.

Proof. Fix two computable copies M and N of $\mathcal{A}_1$.

As usual, we assume that all special points in all copies are pairwise unequal and that the limit point is not special. (If it is, we may non-uniformly remove it from the dense sequence.) As at the very beginning of the proof of Lemma 4.2, we can use scaling by a computable real to make sure that the distances between special points are irrational in both copies. Note both manipulations described above are *ct*-degree invariant, and therefore these properties of M and N can be assumed without any loss of generality. We will also need the following elementary fact.

Lemma 4.5. *In any copy* A *of* $\mathcal{A}_1$*, the unique limit point* $\infty \in A$ *is computable relative to* $0'$*.*

Proof. Since M is computable Polish, it's computably compact relative to $0'$ – i.e., $0'$ can enumerate all finite covers of M by basic open balls. In this way, $0'$ can enumerate clopen partitions of M. (This much relies only on the fact that A is computable Polish and can be found in [11, Lemma 4.2.77].) Moreover, again using only that A is computable Polish, $0'$ can compute if a basic open ball $B(x_k, r)$ contains no special points x_i for $i \geqslant n$:

$$\forall i \geqslant n \forall m \left(|d(x_k, x_i) - q_{i,k,m}| < 2^{-m} \wedge 2^{-m} + q_{i,k,m} < r\right).$$

Since every clopen partition of A has exactly one part which is finite and has arbitrarily small clopen sets around ∞, $0'$ can compute a name for ∞.

□

A rough description of $X \geqslant_{ct} M, N$*.* We build a tree $T \subseteq 2^{<\omega}$ whose right-most path is the characteristic function of K, the halting problem. Our space X will be a superset of $[T]$, where T is considered as a subset of $\omega^{<\omega}$. We will dynamically extend $[T]$ to include isolated paths, at most finitely many at every level of the tree,cin $\omega^\omega \backslash 2^\omega$. In other words, to obtain X we will externally adjoin isolated points to the space $[T]$. The exact number and location of these points will depend on M and N.

The basic strategy. We focus on making sure $X \geqslant_{ct} M$, and we delay the explanation of the combined strategy that will handle both M and N simultaneously.

The construction will be similar to that in the proof of Theorem 3.8, with additional modifications owing to the fact that M and N (and X) now have many isolated points. By Lemma 4.5, there is a functional Φ such that Φ^K computes the limit point ∞ of M. Since K is the right-most path in $[T]$, we can use Φ to guide us in our definition of a reduction $f : X \to M$.

As is now standard, we then build f by linking basic open sets in X with open sets in M and ensuring that the resulting map is well-defined, computable, and bijective.

At each stage s, we will have an approximation σ_s for K, which, via Φ, will give us a guess B as to a small neighbourhood of the limit point ∞ of M. We would like to link $[\sigma_s]$ with B, and declare that $f([\sigma_s]) \subseteq B$. However, if our guess is wrong, and B has fewer points than $[\sigma_s]$, we cannot both satisfy that $f([\sigma_s]) \subseteq B$ and maintain that f is a bijection.

Hence, we do not believe the computation $\Phi^{\sigma_s}(n) = B_s$ until we have seen enough points in B_s for the construction to proceed (possibly by looking ahead in the enumeration of M).

In this way, we can proceed to define an *embedding* with the guidance of Φ, however, it could be not onto. Indeed, maybe very late M shows lots of points outside of B. However, T won't produce any new isolated points outside of $[\sigma]$, which could be an initial segment of $0'$ witnessing a correct Φ-computation. But we have promised that $f([\sigma]) \subseteq B$, and this is certainly an obstacle.

The solution would be to simply put more isolated points, externally, to $[T]\backslash[\sigma]$: simply further extend T by adding external branches at as high a level as possible. Then map these points to the newly enumerated points in $M\backslash B$. The process of adding extra points must stop simply because there are at most finitely many special points outside B. If B or σ are not final, this process also terminates, because there will be a new computation.

For the isolated points, we ensure that there are more points in $X_s\backslash[\sigma_s]$ than in $M_s\backslash B_s$. Extra elements of X_s can be mapped into B_s (by ensuring it is large enough, as in the last paragraph). If there are not enough points in T_s, then we can append new isolated points to X by enumerating a new seuqence τ from $\omega^{<\omega}$. So that X remains compact, we will choose τ in as small of an already-listed open set as possible.

Before we proceed, we also briefly explain why f defined by a procedure outlined above will be a computable homeomorphism. It is trivially computable on the isolated points, and the effective continuity at the limit point is guaranteed by Φ; we commit ourselves to $f([\sigma]) \subseteq B$, and eventually this will be correct. So if we had a name for ∞ in X, we would be able to calculate $f(\infty)$ in M. Also note that f is injective on special points. Since we assumed from the very beginning ∞ is not a special point, the only pre-image of ∞ in X is the right-most path coding K.

The case of two structures. We have outlined a rather general way of producing $X \geqslant_{ct} M$. But of course we also have N. To handle both structures, we run the construction above for *both* both M and N simultaneously. We will be monitor two functionals, Φ and Ψ, and define two maps, f and g. We will require that σ looks good for both computations, with precision 2^{-n}, as made precise in the construction.

Each of the two copies will perhaps force us to add isolated points to our space, as described earlier. Say that B is the current guess for a small neighbourhood of ∞^M and D is the current guess for a small neughbourhood of ∞^N. We will account for new isolated points of X added for the g-strategy by ensuring that there are enough elements of B to "absorb" to the new X-points, and that the X-points are added to an open set that they has not been committed to the complement of B. Indeed we will pick $[\tau]$ for some $\tau \preccurlyeq \sigma$.

Formal proof. We will build a computable enumeration $(T_s)_{s\in\omega}$ of a c.e. tree whose right-most path is the characteristic function of K, and which is closed under appending 0^n for every n. To obtain X, we will append more isolated paths to this tree as M and N are revealed.

We also fix two computable Polish presentations M and N of $\mathcal{A}_1$ which, w.l.o.g., satisfy the properties described at the beginning of the proof:

(1) no repetitions among special points,
(2) distances between the special points are pairwise irrational, and
(3) the limit point is not a special point.

As a consequence of (1) − (3), the dense sequences in our spaces will be composed of exactly all the isolated points, without repetition. Further, by (2), for any special point and any basic open ball (whose radius is rational) we can decide whether the special point is in the ball or outside the respective closed ball. This property will be used in the construction below without explicit reference.

Having in mind Lemma 4.5, non-uniformly fix two operators Φ and Ψ witnessing that the unique limit points in M and N, respectively, are computable relative to the halting problem. (That is, $\Phi^K(n)$ outputs a basic ball in M of radius at most 2^{-n} containing the limit point of M, and similarly for Ψ and N.)

It will be convenient to assume that for every n, the use of $\Phi(n)$ is equal to the use of $\Psi(n)$ and has length at least n. Further, without loss of generality, we can assume that the 2^{-n-1}-ball $\Psi^K(n+1)$ is formally contained in $\Psi^K(n)$.

We also identify $f : X \to M$ with its name consisting of a family of pairs (D, B) so that $f(D) \subseteq B$, and the same for g (see Definition 2.6).

Construction of X. X_s will consist of a subtree $T_s \subseteq 2^{<\omega}$ and a finite set of isolated points $I_s \subseteq \omega^\omega \backslash 2^\omega$. If $\Gamma_s = T_s \sqcup I_s$ and $\Gamma = \bigcup_s \Gamma_s$, then our space X will be equal to the set $[\Gamma]$ of infinite paths through Γ. Noting that Γ will be a c.e. tree with no terminal nodes, this will guarantee that X is computable Polish. The enumeration $(T_s)_{s<\omega}$ will be based on a dynamic approximation to K, as described below.

Stage 0. Set $I_0 := \varnothing$ and $f_0 := g_0 := \varnothing$.
Stage s. Write $ran(f_{s-1})$ for the set of points in M_s that have been *promised* preimages in X_{s-1}, and similarly for N_s and g_{s-1}. Call these sets the *range* of f_{s-1} and g_{s-1} respectively. (This is an abuse of notation, as formally f_{s-1} and g_{s-1} are collections of pairs of open sets, but we will verify at the end that the functions f and g will meet all of our promises made during the construction.)

Search for t such that $\Phi_t^{K_t}\restriction(s+1)\downarrow$ and $\Psi_t^{K_t}\restriction(s+1)\downarrow$ and $\Phi_t^{K_t}(n)$ is a ball of radius at most 2^{-n} for each $n \leqslant s$, and $\Phi_t^{K_t}(n) =: B_n$[1] is a formal subset of $\Phi_t^{K_t}(n+1)$ for $n < s$ (and similarly for $\Psi_t^{K_t}(n) =: D_n$). In addition, we need $M_t \cap B_s$ and $N_t \cap B_s$ to be "large enough": Define

[1]Formally, this should be $B_{n,s}$, but we suppress the second subscript when clear from context.

- k_1 as the number of elements of $M_s \backslash B_s$ that are not in the range of f_{s-1}
- k_2 for the number of elements of $N_s \backslash D_s$ that are not in the range of g_{s-1}
- $\ell = |K_t \backslash K_{t^-}|$, where t^- is "the t" of stage $s-1$ ($t(0) = 0$).

Then t in addition should satisfy that

$$|(M_t \cap B_s) \backslash ran(f_{s-1})| \geqslant k_2 + \ell \quad \text{and} \quad |(N_t \cap D_s) \backslash ran(g_{s-1})| \geqslant k_1 + \ell$$

(Note the difference between the s and t subscripts; here we *look ahead* in the enumerations of M and N to find enough points to proceed with the construction.)

We will verify below that such a t always exists.

Now write σ_s for the use of the computations $\Phi_t^{K_t} \upharpoonright (s+1) \downarrow$ and $\Psi_t^{K_t} \upharpoonright (s+1) \downarrow$ and define T_s to be the downward closure of $T_{s-1} \cup \sigma_s$. We will build I_s, f_s, and g_s as follows.

(1) Enumerate $([\sigma_s], B_s)$ into f_s and $([\sigma_s], D_s)$ into g_s. Thus, we declare that

$$f([\sigma_s]) \subseteq B_s \qquad \text{and} \qquad g([\sigma_s]) \subseteq D_s.$$

(2) Choose $x_i \in M_t \cap B_s$ (resp. $y_j \in N_t \cap B_s$) not in the range of f_{s-1} (resp. g_{s-1}). *Promise* that $f(\sigma_s{}^\frown 0^\omega) = x_i$ and $g(\sigma_s{}^\frown 0^\omega) = x_i$. We take no action towards this promise until the next stage.

(3) For every other $\tau \in T_s \backslash T_{s-1}$, find $x_i \in M_t \cap B_s$ which is not in the range of f_{s-1} and $y_j \in N_t \cap D_s$ that is not in the range of g_{s-1}. *Promise* that $f(\tau^\frown 0^\omega) = x_i$ and $g_{s-1}(\tau^\frown 0^\omega) = y_j$. We will check below that this does not break any earlier promises or declarations.

Formally, we *act* towards this promise by finding a basic open ball V centred at x_i which isolates x_i in M_s. Enumerate $([\tau], V)$ into f_{s-1}. g_{s-1} is built similarly.

(4) For each $x_i \in M_s \backslash B_s$ with $x_i \notin ran(f_{s-1})$, let $\tilde{B}$ be the least basic open set containing x_i and for which we have declared that $f([\tau]) \subseteq \tilde{B}$ for τ on the rightmost path of T_s. Then, by our conventions on X_s, $\tilde{B} \supseteq B$. Let σ_* be the least extension of τ on the rightmost path of T_s with $\sigma_* = \sigma_t$ for some t. Furthermore, we have that there is $\tilde{D} \supseteq D_s$ with $g_{s-1}([\sigma_s]) \subseteq \tilde{D}$. By assumption on the size of $N_t \cap D_s$, for each such x_i, there is $y_{j_i} \in N_t \cap D_s$ which is not in the range of g_{s-1}, nor was picked at an earlier substage.

Enumerate a new point $\rho \in \omega^{<\omega} \backslash I_s$ with $|\sigma_*^-| = |\rho|$ into I_s. *Promise* that $f(\rho^\frown 0^\omega) = x_i$ and $g(\rho^\frown 0^\omega) = y_{j_i}$, and act towards the promise.

(5) Perform the analogue of (2) for each $y_j \in N_s \backslash D_s$.

(6) Now, we act towards promises made at earlier stages. For each $x_i \in M_s \backslash B_s$, there is $\rho_i \in T_s \cup I_s$ such that we have promised $f(\rho_i{}^\frown 0^\omega) = x_i$. Let $(U, V) \in f_{s-1}$ be minimal w.r.t. inclusion with $x_i \in U$ and $\rho_i{}^\frown 0^\omega \in V$. Search for $\tilde{U} \ni x_i$ and $\tilde{V} \ni \rho_i{}^\frown 0^\omega$ with $\tilde{U} \subset U$ and $\tilde{V} \subset V$, radii $< 2^{-s}$, and $\tilde{U}$ isolates x_i in M_s and $\tilde{V}$ isolates $\rho_i{}^\frown 0^\omega$ in X_s. Enumerate $(\tilde{U}, \tilde{V})$ into f_s.

(7) Perform the analogue of (4) for $y_j \in N_s \backslash D_s$.

(8) Set $\Gamma_s = T_s \cup I_s$ and $X_s = \{\tau^\frown 0^\omega \mid \tau \in \Gamma_s\}$. Set f_s to to be superset of f_{s-1} obtained by adding the pairs of open sets from (1), (2), and (4); while g_s is the superset of g_{s-1} obtained by the addition of the pairs of open sets from (1), (3), and (5).

Proceed to stage $s+1$.

This concludes the construction.

Verification.

Lemma 4.6. *For every s, stage s finishes its work.*

Proof. We first need to argue that a stage t is found satisfying

(1) $\Phi_t^{K_t}\upharpoonright(s+1)\downarrow$;
(2) $\Psi_t^{K_t}\upharpoonright(s+1)\downarrow$;
(3) $\Phi_t^{K_t}(n)$ is a ball of radius at most 2^{-n} for each $n \leqslant s$, and $\Phi_t^{K_t}(n)$ is a formal subset of $\Phi_t^{K_t}(n+1)$ for $n < s$;
(4) $\Psi_t^{K_t}(n)$ is a ball of radius at most 2^{-n} for each $n \leqslant s$, and $\Psi_t^{K_t}(n)$ is a formal subset of $\Psi_t^{K_t}(n+1)$ for $n < s$;
(5) $|(M_t \cap B_s)\backslash ran(f_{s-1})| \geqslant |(N_s\backslash D_s)\backslash ran(g_{s-1})| + \ell$; and
(6) $|(N_t \cap D_s)\backslash ran(g_{s-1})| \geqslant |(M_s\backslash B_s)\backslash ran(f_{s-1})| + \ell$.

Notice that since Φ^K computes a fast Cauchy sequence for the limit point of M, there is t' such that $\Phi_{t'}^{K_{t'}}$ satisfies conditions (1) and (3) and $K_{t'} = K$ on the use of the computation in (1). Since $\Phi_{t'}^{K_{t'}}(s)$ contains the limit point of M, it also contains infinitely many points of M. Thus there is $t'' \geqslant t'$ satisfying conditions (1), (3), and (5). Arguing similarly for Ψ and N, we obtain $t''' \leqslant t''''$. t is the maximum of t'' and t''''.

It is not hard to check that every other search performed at stage s is bounded. □

Lemma 4.7. *The names of f and g constructed by the procedure are names of computable bijections $X \to M$ and $X \to N$.*

Proof. We prove the lemma for f, the proof for g is the same. Write σ for the characteristic function of K.

We show, in turn, that each isolated point in X is promised to an isolated point in M; that each special point in M is promised to some isolated point in X; that all promises are kept; that f is well-defined; and that $f(\sigma)$ is the limit point of M.

Let $\rho^\frown 0^\omega$ be an isolated point in X, and that $\rho^\frown 0^\omega$ enters X at stage s. (Note that every isolated point in X has this form.) If $\rho \notin 2^{<\omega}$, then, at the stage ρ is added, ρ is promised to a special point in M (substage (4) or (5)). During substage (5), we use that there are at least $k_2 + \ell$ elements of $M_t \cap B_s$ to find enough new, un-promised isolated points of M to be images of the new isolated points of X. If $\rho \in 2^{<\omega}$, then $\rho \preccurlyeq K_t$ for some approximation to K satisfying the conditions laid out at the beginning on stage s (and in the previous lemma). Then ρ is promised to x_i at substage (2), if $\rho = \sigma_s$, or at substage (3), if $\rho \neq \sigma_s$. Here we use that there are at least ℓ isolated points in $M_t \cap B_s$ to find the necessary images for the isolated points of X.

Let x_i be an isolated point in M, and suppose that x_i becomes visible to the construction at stage s – i.e., either $x_i \in M_s$ or $x_i \in M_t \cap B_s$. If $x_i \in M_s\backslash B_s$, then it is promised a pre-image in substage (4). If $x_i \in M_t \cap B_s$, then we do not necessarily find a pre-image for it at stage s. If it is promised to an element of X during substages (2), (3), or (7), we are done. Suppose that x_i is never chosen during those substages. Then, since x_i is isolated, there will be a stage $v \geqslant s$ at which $x_i \notin B_v$. At stage v, substage (4), x_i will be promised a pre-image.

Now we show that f is a well-defined computable homeomorphism. This implies that if $\rho \in X$ (note that ρ denotes an infinite sequence) is promised to x_i, then $f(\rho) = x_i$, by the action of substage (6).

We list a few observations about the construction that will be useful in the argument.

- If $\rho \not\preccurlyeq \sigma_s$ and U isolates ρ in X_s, then U isolates ρ in X.
- The only instance of $(U, V) \in f$ with U not isolating an element of X is when $U = [\sigma_s]$ and $V = B_s$.
- No action is taken on a promise of the form "$f(\sigma_s{}^\frown 0^\omega) = x_i$" until stage $s+1$, at which point $\sigma_{s+1} \neq \sigma_s$, so we can find an isolating set for $\sigma_s{}^\frown 0^\omega$.

Thus, if (U, V) is enumerated into f_s as part of an action towards keeping the promise $f(\rho) = x_i$, then U is an isolating set for ρ in X_t for every $t \geqslant s$. By the actions of substage (6), sets (U_t, V_t) are enumerated into f_t with $\rho \in U_t$, $x_i \in V_t$, and the radii of U_t and V_t both $\leqslant 2^{-t}$. Hence $f(\rho) = x_i$.

On the other hand, if $U = [\sigma_s]$ and $V = B_s$, let $\rho \in [\sigma_s]$. If $\rho = \sigma_s{}^\frown 0^\omega$, then ρ is promisesd to $x_i \in B_s$ at stage s. At stage $s+1$, open sets U and V are found to act towards this promise. By the previous paragraph, $f(\rho) = x_i \in B_s$, as required. If $\rho = \tau^\frown 0^\omega$ for some τ properly extending σ_s, then $\Phi_t^\tau(s) = B_{s,s}$ for every $t \geqslant s$. Thus, when $\rho^\frown 0^\omega$ is promised some image x_i, it will be in some $B_v \subseteq B_s$. Finally, if $\rho = \sigma$, then Φ^{σ_s} is a correct computation, so the limit point of M really is in B_s. Furthermore, $\sigma_s \preccurlyeq \sigma_t$ for every $t \geqslant s$, so $\Phi_t^{\sigma_t}(t) \subseteq B_s$. Hence, via the work of substage (1) during every stage $t \geqslant s$, we guarantee that $f(\rho)$ is the limit point of M.

Implicit in the preceding argument is that for every $x \in M$ and every n, there is a stage s and a basic open $U \ni x$, and a basic open $V \ni f(x)$ with radius $\leqslant 2^{-n}$ such that at stage s, we declare that $f(U) \subseteq V$.

Since the construction is evidently computable, f is a well-defined computable homeomorphism from $X \to M$. □

□

5. Proof of Theorems 1.5 and 1.7

Recall that Theorems 1.5 and 1.7 show that topological properties of Stone spaces are equivalent to certain properties of the *ct*-degrees on the space. In particular, Theorem 1.5 states that, for a computable Stone space S, S has finitely many limit points iff every copy of S is *ct*-above a computably compact copy. Since we don't know, even for Stone spaces, whether there can be *ct*-minimal structures that are not computably compact, this is not an order-theoretic property of the space, it nonetheless entails Theorem 1.7 that a Stone space S is finite iff $\mathbf{CT}(S)$ has a minimum degree.

We write $L \cdot L_0$ for the product of two linear orders L and L_0; we shall interpret it as every point in L_0 being "replaced" with an interval isomorphic to L (this is not to be confused with the lexicographic product which is symmetric to this). So, for example, if η is the order of the rationals, $2 \cdot \eta$ is a countable family of 2-element orders put together densely.

Before we prove Theorems 1.5 and 1.7, we start with a lemma.

Lemma 5.1. *Suppose a computable Stone space S contains a clopen subset homeomorphic to one of the following spaces:*

(1) $\mathcal{B} \cong 2^\omega$,
(2) $\mathcal{A}_2 \cong \widehat{Intalg(\omega^2)}$,

(3) $\mathcal{B}_1 \cong \widehat{Intalg(2 \cdot \eta)} \cong \widehat{Intalg(\omega \cdot \eta)}$,

(4) $\mathcal{B}_2 \cong \widehat{Intalg((\omega + \omega) \cdot \eta)} \cong \widehat{Intalg(\omega^2 \cdot \eta)}$.

Then S has a ct-degree which is not ct-above any computable compact degree.

Furthermore, in each case, there is a c.e. closed subset of $[0, 1]$ *not ct-above any computably compact copy (cf. Theorem 4.1).*

Extended sketch. In all four cases, we will imitate the proof of Theorem 3.1, with only minimal modifications in (2), and with insignificant modifications in (1), (3) and (4). In all cases, the argument will work by running the construction of Theorem 3.1 on the given clopen set.

This restriction will be done as follows. Using Theorem 2.9, fix a computably compact presentation S_0 of S. We can *non-uniformly* fix finitely many basic clopen balls whose union is equal to the clopen component D of S_0 having the described form (as described in (1), (2), (3), or (4)). To build our space M not *ct*-above any computably compact copy, we will keep $S_0 \backslash D$ unchanged, but we will replace D with a new clopen component (which also internally has a new compatible metric) homeomorphic to the original D. We will keep the distance between $S_0 \backslash D$ and D the same as before, and we will scale down the new metric inside the new copy of D if necessary (to keep its diameter of the component as small as it was before the replacement). Since this can all be done computably, in what follows, we will just describe how to meet all of the requirements for not being *ct*-above a computably compact copy for the spaces $\mathcal{A}_2$, $\mathcal{B}_2$, and $\mathcal{B}_1$.

Recall that the initial observation in the proof of Theorem 3.1 is the following property, which we call (*):

(*) *If $M \geqslant_{ct} S_0$, then there is a c.e. list of clopen partitions of M – i.e., a c.e. collection of pairs of clopen sets (U_i, V_i) such $M = U_i \sqcup V_i$ and any two distinct points of M are separated by some partition. (Here U_i and V_i are each represented as a finite union of basic open balls in M.)*

For each potential list W_e of such clopen partitions, we built $M_e \cong 2^\omega$ on which (*) fails for W_e by, in the infinitary outcome, built a series of test points limiting to the boundary between U_i and V_i. As we added these test points, we built small copies of Cantor space around them. Since the disjoint union of finitely copies of Cantor space is again homeomorphic to Cantor space, and the one point compactification of countably many copies of Cantor space is homeomorphic to Cantor space, the result is a copy of Cantor space. The final space M is the one-point compactification of $\bigsqcup_e M_e$, and so is homeomorphic to Cantor space.

By [11, Lemma 4.2.77], any computably compact copy of a computable Polish space admits a c.e. enumeration of its clopen partitions. Thus, in each of the cases (1)–(4), it suffices to diagonalise against (*).

The proof for $S \cong \widehat{Intalg(\omega^2)}$ is very similar to the proof of Theorem 3.1. We implement a similar basic strategy, but this time we will not introduce a small copy of 2^ω around the test points every time we attempt to diagonalise. As a result, our space will be a “fishbone” of spaces M_i, where each M_i is either finite of has exactly one limit point, depending on whether the outcome is finitary or infinitary, respectively. Of course, we will have infinitely many infinite M_i. (Clearly, we could also ensure this directly too, by introducing an infinite subsequence in M_i of copies of $\mathcal{A}_1$ which are not even involved in the diagonalisation.) Thus, the resulting space is homeomorphic to $\widehat{Intalg(\omega^2)}$, since again, the one point compactification of $\bigsqcup_i M_i$, where each $M_i \cong \mathcal{A}_1$ is exactly $\mathcal{A}_2$.

The proof of (3) resembles Theorem 3.1 more closely that when $S \cong \widehat{Intalg(\omega^2)}$. This time, we repeat the proof of Theorem 3.1 essentially verbatim, but replace every instance of $\mathcal{B} \cong 2^\omega \cong \widehat{Intalg(\eta)}$ with $\mathcal{B}_1 \cong \widehat{Intalg(2 \cdot \eta)}$.

As long as the homeomorphism class of $\mathcal{B}_1$ is closed under taking finite disjoint unions and under the operation of taking the 1-point compactification of a countable disjoint union, the result will be homeomorphic to $\mathcal{B}_1$. This follows, after applying Stone duality, from the algebraic characterisation of $\mathcal{B}_1$ that can be found in [4, Proposition 2]. (For a more direct argument, consider $\widehat{Intalg(L)}$, where $L = \sum_{i=1\ldots,k} \mathcal{B}_1$ and $L = \sum_{i\leqslant 0; i\in\mathbb{Z}} \mathcal{B}_1 = \hat{\mathcal{B}}_1 \cdot \omega^*$, representing the disjoint union of finitely many copies of $\mathcal{B}_1$ and the one-point compactification of $\mathcal{B}_1$, respectively. Alternatively, note that $\mathcal{B}_1$ can be represented as $[\Gamma]$, where Γ is a ternary tree obtained from the complete binary tree $2^{<\omega}$ by adjoining an extra isolated path to every node of $2^{<\omega}$. In each case, use a back-and-forth argument taking into account the isolated points.)

This gives (3).

Case (4) is essentially the same as (3), taking into account that the homeomorphism class of $\mathcal{B}_2$ is also closed under finite disjoint unions and one-point compactifications. This follows from the characterisation of $\mathcal{B}_2$ in [4, Proposition 5], or by a back-and-forth argument as hinted at above. In this case, consider the representation of $\mathcal{B}_2$ as $[\Theta]$, where Θ is the ternary tree obtained from appending a fishbone to each node of $2^{<\omega}$.

In each case, since $S_0 \backslash D$ was in fact computably compact, we can realise the resulting space as a c.e. closed subspace of $[0, 1]$. □

We of course realise that $(1), (3)$, and (4) are clearly consequences of a more general result. With some effort, this general result can perhaps be formulated to cover (2) too. However, we will leave the exact formulation of this result to the reader.

Write $(M)'_{CB}$ for the Cantor-Bendixson derivative of M. Recall that Theorem 1.5 states that for a (computable) Stone space S, the following conditions are equivalent:

(1) Every *tc*-degree of S bounds a computably compact (thus, minimal) degree.
(2) $(S)'_{CB}$ is finite.

Proof of Theorem 1.5. The case when S is finite is trivial. The case when S is infinite and $(S)'_{CB}$ is finite follows from Theorem 4.1. Indeed, if there are finitely many limit points, then (non-uniformly) split the space into finitely many disjoint clopen sets $(M_i)_{i\leqslant n}$, each homeomorphic to $\mathcal{A}_1$, and each represented as a finite union of basic open sets. Then Theorem 4.1 produces a *ct* reduction $f_i : M_i \to [0, 1]$ such that $ran(f_i)$ is computably compact. Since the union of finitely many computably compact spaces is computably compact (and the union of finitely many disjoint copies of the unit interval is computably homeomorphic to the unit interval), the union of these maps is a *ct*-reduction from S to a computably compact copy of S.

Now, assume S has infinitely many limit points, and suppose S *ct*-bounds a computably compact copy.

Then Lemma 5.1(2) guarantees that, further, $(S)''_{CB}$ has no isolated points. If $(S)''_{CB} = \varnothing$, then (by compactness) $(S)'_{CB}$ is finite, contradicting our hypothesis.

Thus, $(S)''_{CB}$ is nonempty with no isolated points, so we may assume $(S)''_{CB} \cong 2^\omega$ throughout. By Lemma 5.1(1), no clopen set of S is homeomorphic to 2^ω. Similarly, by Lemma 5.1(3), no clopen set is homeomorphic to $\mathcal{B}_1$. Thus, if a clopen subset D of S is infinite, then has to contain a clopen copy of $\mathcal{A}_1$: by compactness it contains a limit point, and if no clopen subset were homeomorphic to $\mathcal{A}_1$, then it could contain a perfect set of limit points. Since $(S)''_{CB} \cong 2^\omega$, it would then have to contain a copy of 2^ω or $\mathcal{B}_1$, contradicting the argument above.

Thus, depending on terminology, the dual Boolean algebra is "1-atomic" or "2-atomic", meaning that every element whose principal ideal is infinite bounds a 1-atom.

It follows that we can split S into disjoint clopen sets S_0, S_1 and $D_\varepsilon \cong \mathcal{A}_1$, where S_0 and S_1 are both infinite and contain infinitely many copies of $\mathcal{A}_1$. We can iterate this process to define S_σ and D_σ, $\sigma \in 2^\omega$. Since S contains no copy of $\mathcal{A}_2$, any clopen set with infinitely many copies of $\mathcal{A}_1$ admits a partition into two clopen components each having infinitely many copies of $\mathcal{A}_1$. Thus, we will never reach S_σ which cannot be thus refined. This property can be used to run a back-and-forth procedure witnessing that $S \cong \mathcal{B}_2$. This material is considered folklore; the more general fact is [16, Exercise 3, Section 1.5], and the case of $\mathcal{B}_2$ is explicitly stated in [4, Proposition 5]. (Again, one can alternatively appeal to the tree representation of $\mathcal{B}_1$ discussed in the previous proof.)

However, Lemma 5.1(4) rules out this possibility.

Hence, S has a copy not *ct*-above a computably compact copy, and Theorem 1.5 is proved.

□

We finish the section by deducing Theorem 1.7 from the previous result. Recall that Theorem 1.7 states that, for a (computable) Stone space S, the following are equivalent:

(1) S is finite;
(2) $\mathbf{CT}(S)$ has a least degree.

Proof of Theorem 1.7. If S is finite, then all of its computable presentations are trivially computably compact, and all lie in the same degree. Thus, assume S is infinite. Every computable Stone space has a computably compact degree, by Theorem 2.9, and it has to be minimal. In particular, if there is a least degree, it has to be computably compact. By Theorem 1.5, if S has a least degree, then S' needs to be finite. In this case, since S is infinite, it has to be a disjoint union of copies of $\mathcal{A}_1$. However, in this case S has infinitely many incomparable computably compact (thus, minimal) degrees by Corollary 2.11, and thus no least degree. □

6. Upper bounds. Proof of Theorem 1.8

In this section we characterise when two copies have an upper bound. Notice that this gives the first example of a first-order difference between the posets $\mathbf{CT}(\mathcal{A})$ and $\mathbf{CT}(\mathcal{B})$ for $\mathcal{A}$ and $\mathcal{B}$ not homeomorphic and infinite.

Lemma 6.1. *For each of the following Stone spaces, there exist a pair of computable Polish presentations of the space without a common upper bound under* $\leqslant_{ct}$*:*

(1) $\mathcal{A}_2 \cong \widehat{Intalg(\omega^2)}$,
(2) $\mathcal{B}_1 \cong \widehat{Intalg(2 \cdot \eta)}$,
(3) $\mathcal{B}_2 \cong \widehat{Intalg((\omega + \omega) \cdot \eta)}$.

Furthermore, the same can be said about any Stone space having a clopen subset homeomorphic to either of these spaces in (1)–(3).

We say that in a computable Polish space *isolated points are c.e.* if there exists a c.e. set of basic open balls $(B_i)_{i\in\omega}$ so that each B_i contains only its center point (i.e., "isolates" its center) and furthermore every isolated point of the space is a center of one of these balls. The proof is based on the following observation.

Observation 6.2. If the isolated points in M are c.e. and $N \geqslant_{ct} M$ then the isolated points of N are c.e.

Indeed, if B isolates its centre, then $f^{-1}(B)$ is equal to the first ball it enumerates. Hence if (B_i) witnesses that the isolated points of M are c.e., then $(f^{-1}(B_i))$ witnesses that the isolated points on N are c.e.

Proof of Lemma 6.1. Observe that the spaces from (1)–(3) all have copies in which isolated points are c.e. It is therefore sufficient to produce a copy M of such a space so that in *any* other $A \geqslant_{ct} M$ the set of isolated points is not c.e. Then, by Observation 6.2, this copy M and the "natural" copy with c.e. isolated points cannot possibly have a common upper bound.

The proof of the lemma will strongly resemble the proof of Theorem 3.1. We construct a 'fishbone space' M, which is a 1-point compactification of the disjoin sequence of spaces (M_i). The ith requirement is responsible for diagonalising against the ith triple of the form (A, f, W), where A is a (potential) computable Polish presentation of our space, f witnesses the potential reduction, and W attempts to list all isolated points in A.

For simplicity of exposition, in this construction, it will be useful to consider f as mapping fast Cauchy names to fast Cauchy names, rather than as an enumeration operator listing open balls mapping into open balls. (See [11, Lemma 2.3.13].)

We look at one such $R_i = (A, f, W)$ in isolation. In all three cases (1)–(3), the blueprint of our strategy is as follows.

(1) Wait for a basic open $B(x, r)$ to be listed in W with $f(x) \in D$ where D has small radius (to be picked during the construction) and is disjoint from the balls associated to higher priority requirements.
(2) If $f(x)$ is well-defined, we will turn it into a limit point by adding a converging sequence to D.

The choice of the radius of D and its location will allow us to put a restraint on parts of our space in presence of many strategies; we will give more details later. The strategy of turning $f(x)$ into a limit point will depend on the homeomorphism type of the space that we construct.

Case 1: Building $M \cong \mathcal{A}_2$.

Fix the priority ordering $R_0 < R_1 < \cdots$ on the requirements R_i discussed above.

Mimicking the set-up of the proof of Theorem 3.1, we build a fishbone of copies of $\mathcal{A}_1$. We refer to the *component* $[0^e\hat{}1]$ as M_e.
Strategy for $R_i = (A, f, W)$: Wait for $B(x, r) \in W$ such that $f(x)$ is guaranteed to be in a single component M_e and M_e is not associated to a higher priority requirement. Associate M_e to R_i and, if M_e is associateed to a lower priority requirement, initialise that requirement.

Notice that if W is an enumeration of isolating balls of all isolated points of A and f is a well-defined, onto homeomorphism, then this search must eventually halt at some finite stage. At this stage, stop building $\mathcal{A}_1$, and proceed to diagonalise as follows.

Let $f(x) = (a_i)$ be the fast Cauchy name that f outputs given a fast Cauchy name for x. Add a point y to B, which we will ensure is isolated, unless $f(x) = y$. (For example, we can assume that we are building the copy of $\mathcal{A}_1$ at M_e inside the Hilbert cube H. At stage s, we ensure that $M_{e,s}$ is a subset of the first s dimensions of H.

If at any stage we see that f is inconsistent; e.g. if a_i and a_{i+1} are too far apart, then we proceed to build a copy of $\mathcal{A}_1$.
Construction: At stage s, perform the strategy for R_i, $i \leqslant s$, in priority order, as described above. Moreover, for each M_e, $e \leqslant s$ not associated to a requirement, extend M_e to a canonical s-element approximation to $\mathcal{A}_1$.
Verification: We show by induction that each R_i is associated to a fixed M_e cofinitely often, or is waiting cofinitely often. Suppose this is true for every R_j, $i < i$. Let s be a stage after which each R_j, $j < i$ has settled. Either R_i is waiting from stage s onwards, in which case we are done, or R_i is eventually associated with some M_e. By assumption, no higher priority requirement injures this association, so R_i remains associated to M_e, as required. Moreover, the requirements associated to a fixed M_e are weakly increasing in priority strength, so if M_e is associated to a requirement infinitely often, then it must be associateed to a fixed R_i cofinitely often.

Thus each $M_e \cong \mathcal{A}_1$: if M_e is cofinitely associated to R_i, then the stratgy for R_i ensures that $M_e \cong \mathcal{A}_1$. Otherwise, M_e is cofinitely unassociated cofinitely often. Hence the construction guarantees that $M_e \cong \mathcal{A}_1$.

Finally, we check that each $R_i = (A, f, W)$ is met. Indeed, if R_i waits forever, or if we see that f is inconsistent, then W does not list all the isolated points or f fails to be a computable homeomorphism. Else, the strategy for R_i guarantees that $f(x)$ is a limit point. Hence, either f fails to be a homeomorphism or W fails to list only the isolated points.

Case 2: Building $M \cong \mathcal{B}_1$. This is very similar to Case 1, so we provide less detail. Recall that $\mathcal{B}_1$ can be represented as the tree obtained by adding an isolated point extending every node of $2^{<\omega}$. The key difference from Case 1 is that this time we will construct a copy of $\mathcal{B}_1$ "around" $f(x)$ (more precisely, around a_i). This is done using the method from the proof of Theorem 3.1. It is clear that $\mathcal{B}_1$ has a 'natural' representation of the form $[T]$, where the right-most path of T is a computable limit point ξ. (Compare to $C(A)$ from Theorem 3.6.)

We again start with a copy of the fishbone.

The strategy for R_i again is to wait for $f(x)$ to map into a component M_e not associated to a higher priority R_j. Say that $f_s(x{\upharpoonright}s) = a_i$ and $(a_i - 2^{-i}, a_i + 2^{-i}) \subseteq$

M_e. Commence construction of a small copy of $\mathcal{B}_1$ in a $2^{-(i+1)}$-interval J_i around a_i in M_e. As discussed above, we can ensure that this copy of $\mathcal{B}_1$ has a_i as its right-most element. While we wait for f to give us a_{i+1}, continue builgin this copy of $\mathcal{B}_1$. Once, if ever, a_{i+1} is defined, choose a $2^{-(i+2)}$-interval J_{i+1} around a_{i+1} and repeat the process inside J_{i+1}. In addition, begin turning all points in $D \backslash J_{i+1}$ into disjoint copies of $\mathcal{B}_1$. The rest is the same as in Case 1 described earlier, and $M \cong \mathcal{B}_1$ since, again, it is either a finite disjoint union of copies of $\mathcal{B}_1$ or the 1-point compactification of an infinite disjoint union of copies of $\mathcal{B}_1$.

Case 3: Building $M \cong \mathcal{B}_2$. Same as Case 2, with $\mathcal{B}_1$ replaced with $\mathcal{B}_2$ throughout.

Finally, note that in all three cases the strategies were localised to D and M_i, and used that every surjective f has to produce a pair of witnesses x and $D \ni f(x)$ with D arbitrarily small and located in any clopen subset of the space of our choice. In other words, we can localise the strategy to a clopen set of the space homeomorphic to one of the spaces $\mathcal{A}_1$, $\mathcal{B}_1$, or $\mathcal{B}_2$, as explained in the proof of Lemma 5.1. □

6.1. **A general condition.** A Boolean algebra is atomic if every element of the algebra bounds an atom, in other words, no element is atomless. Under the duality, it is equivalent to saying that any clopen set contains an isolated point. We write B' to denote the Boolean algebra of the Stone space $(\widehat{B})'_{CB}$.

Theorem 6.3. *Let B be an atomic Boolean algebra so that B' is infinite. Suppose that B has a computable presentation in which the set of atoms is computable too. Then $\mathbf{CT}(\widehat{B})$ has a pair of degrees with no common upper bound.*

Proof. Under Stone duality, atoms correspond to the collection of isolated clopen neighbourhoods in some canonical copy of $S = \widehat{B}$ given as $[T]$; cf. Lemma 6.2. So by Claim 6.2 any copy above $S = [T]$ will have c.e. isolated points.

The proof now splits into cases closely resembling those in the proof of Theorem 1.5. If there is a clopen set of the form $\mathcal{A}_2$, then the theorem follows from Lemma 6.1(1). Otherwise, $(S)''_{CB}$ is either 2^ω or empty. However, by our assumption $(S)'_{CB}$ is infinite, and thus by compactness $(S)''_{CB}$ cannot be empty. Thus, $(S)''_{CB}$ is 2^ω. Note that B is atomic, and therefore S contains no clopen set homeomorphic to 2^ω. If it has a clopen subset with no clopen copy of $\mathcal{A}_1$ in it, then this clopen set must be a copy of $\mathcal{B}_1$. Indeed, its Boolean algebra is atomic, infinite, and its factor by the ideal generated by the atoms is infinite and atomless. These properties describe $\mathcal{B}_1$ uniquely up to isomorphism, e.g., [4, Proposition 2]. In this case we apply Lemma 6.1(2).

The remaining case is when every clopen subset of S has a clopen copy of $\mathcal{A}_1$ in it (together will all the earlier assumptions). Just as in the proof of Theorem 1.5, in this case we have $S \cong \mathcal{B}_2$. □

6.2. **Proof of Theorem 1.8.** Suppose $S = \widehat{B}$ is a computable Stone space with trivial perfect kernel. We need to prove that the following conditions are equivalent:

(1) Every pair of degrees in $\mathbf{CT}(S)$ has a common upper bound.
(2) $(S)'_{CB}$ is finite.

If $(S)'_{CB}$ is finite, then either S is finite or it is a finite union of clopen homeomorphic copies of $\mathcal{A}_1$. In the former case, all copies are bi-computably homeomorphic, so $\mathbf{CT}(S)$ is a single point. In the latter case, consider computable copies

$A = A_0 \sqcup \cdots \sqcup A_{n-1}$ and $B = B_0 \sqcup \cdots \sqcup B_{n-1}$ of S where $a_i, B_i \cong \mathcal{A}_1$. We apply Theorem 4.4 to produce an upper bound C_i for A_i and B_i. Then the upper bound of A and B is $C_0 \sqcup \cdots \sqcup C_{n-1}$.

If $(S)'_{CB}$ is infinite, then we claim that the result follows from Theorem 6.3. Indeed, it is well-known that the Boolean algebra $B \cong \widehat{S}$ has to be of the form $Intalg(\alpha)$ for some computable ordinal α. Boolean algebras of this form are called *superatomic* in the literature; e.g., [16]. By Theorem 2.9, B is computable. Superatomic Boolean algebras have decidable presentations (e.g., [16, Corollary 3.2.1]), and in such copies the set of atoms is computable. It remains to apply Theorem 6.3.

References

[1] N. Bazhenov, M. Harrison-Trainor, and A. Melnikov. Computable Stone spaces. *Ann. Pure Appl. Logic*, 174(9):Paper No. 103304, 2023.

[2] N. Bazhenov, I. Kalimullin, A. Melnikov, and K. M. Ng. Online presentations of finitely generated structures. *Theoret. Comput. Sci.*, 844:195–216, 2020.

[3] N. Bazhenov, A. Melnikov, and K. M. Ng. Every Δ^0_2 Polish space is computable topological. *Proc. Amer. Math. Soc.*, 152(7):3123–3136, 2024.

[4] N. A. Bazhenov. Δ^0_2-categoricity of Boolean algebras. *Journal of Mathematical Sciences*, 203(4):389–400, 2014.

[5] Nikolay Bazhenov, Matthew Harrison-Trainor, and Alexander Melnikov. Computable stone spaces. *Annals of Pure and Applied Logic*, 174(9):103304, 2023.

[6] Volker Bosserhoff and Peter Hertling. Effective subsets under homeomorphisms of $\mathbb{R}^n$. *Inform. and Comput.*, 245:197–212, 2015.

[7] Vasco Brattka and Gero Presser. Computability on subsets of metric spaces. *Theoretical Computer Science*, 305(1-3):43–76, 2003.

[8] G. S. Ceitin. Algorithmic operations in construcive complete separable metric spaces. *Doklady Akademii Nauk.*, 128:49–52, 1959.

[9] Peter Cholak, S. Goncharov, Bakhadyr Khoussainov, and Richard Shore. *Journal of Symbolic Logic*, 64:13–37, 1999.

[10] B. Csima and J. Stephenson. Finite computable dimension and degrees of categoricity. *Annals of Pure and Applied Logic*, 170(1):58–94, 2019.

[11] R. Downey and A. Melnikov. *Computable Structure Theory: A Unified Approach (in print).*

[12] R. Downey and A. Melnikov. Computably compact metric spaces. *The Bulletin of Symbolic Logic*, 29(2):170–263, 2023.

[13] S. Goncharov. Computable single-valued numerations. *Algebra and Logic*, 19:325–356, 1980.

[14] S. Goncharov. The problem of the number of nonautoequivalent constructivizations. *Algebra i Logika*, 19(6):621–639, 745, 1980.

[15] S. Goncharov. Limit equivalent constructivizations. In *Mathematical logic and the theory of algorithms*, volume 2 of *Trudy Inst. Mat.*, pages 4–12. "Nauka" Sibirsk. Otdel., Novosibirsk, 1982.

[16] S. Goncharov. *Countable Boolean algebras and decidability.* Siberian School of Algebra and Logic. Consultants Bureau, New York, 1997.

[17] S. Goncharov and V. Dzgoev. On constructivizations of some structures. *Akad. Nauk SSSR, Sibirsk. Otdel., Novosibirsk, 1978*, 1978. (manuscript deposited at VINITI on July 26, 1978).

[18] S. Goncharov and V. Dzgoev. Autostability of models. *Algebra i Logika*, 19(1):45–58, 132, 1980.

[19] S. Goncharov, S. Lempp, and R. Solomon. The computable dimension of ordered abelian groups. *Adv. Math.*, 175(1):102–143, 2003.

[20] N. Greenberg, Matthew Harrison-Trainor, A. Melnikov, and D. Turetsky. Non-density in punctual computability. *Ann. Pure Appl. Logic*, 172(9):Paper No. 102985, 17, 2021.

[21] N. Greenberg, J. F. Knight, A. Melnikov, and D. Turetsky. Uniform procedures in uncountable structures. *J. Symb. Log.*, 83(2):529–550, 2018.

[22] Vassilios Gregoriades, Tamás Kispéter, and Arno Pauly. A comparison of concepts from computable analysis and effective descriptive set theory. *Math. Structures Comput. Sci.*, 27(8):1414–1436, 2017.

[23] M. Harrison-Trainor, A. Melnikov, and K. M. Ng. Computability of Polish spaces up to homeomorphism. *The Journal of Symbolic Logic*, 85(4):1664–1686, 2020.
[24] Matthew Harrison-Trainor, A. Melnikov, and A. Montalbán. Independence in computable algebra. *J. Algebra*, 443:441–468, 2015.
[25] M. Hoyrup, T. Kihara, and V. Selivanov. Degree spectra of homeomorphism type of compact Polish spaces. *J. Symbolic Logic*, 2023. available online.
[26] Zvonko Iljazović. Isometries and computability structures. *J.UCS*, 16(18):2569–2596, 2010.
[27] Zvonko Iljazović and Takayuki Kihara. Computability of subsets of metric spaces. In *Handbook of computability and complexity in analysis*, Theory Appl. Comput., pages 29–69. Springer, Cham, 2021.
[28] I. S. Kalimullin, A. G. Melnikov, and K. M. Ng. The diversity of categoricity without delay. *Algebra Logic*, 56:171–177, 2017.
[29] Kai Jun Khoo, Heer Tern Koh, and Keng Meng Ng. A discrete linear order with non-dense punctual degrees. *Theoretical Computer Science*, 1047:115324, 2025.
[30] H. T. Koh, A. Melnikov, and K. M. Ng. Counterexamples in effective topology. To appear in JSL.
[31] Heer Tern Koh and Keng Meng Ng. The punctual degrees of equivalence relations. *Submitted for Publication*, 2024.
[32] P. LaRoche. Recursively presented Boolean algebras. *Notices AMS*, 24:552–553, 1977.
[33] Oscar Levin. Computable dimension for ordered fields. *Arch. Math. Logic*, 55(3-4):519–534, 2016.
[34] M. Lupini, A. Melnikov, and A. Nies. Computable topological abelian groups. *J. Algebra*, 615:278–327, 2023.
[35] A. Mal'cev. Constructive algebras. I. *Uspehi Mat. Nauk*, 16(3 (99)):3–60, 1961.
[36] T. McNicholl. Computable copies of ℓ^p. *Computability*, 6(4):391–408, 2017.
[37] Timothy H. McNicholl. Evaluative presentations. *Journal of Logic and Computation*, 35(5):exaf036, 2025. Open Access; published 19 June 2025.
[38] A. Melnikov. Computably isometric spaces. *J. Symbolic Logic*, 78(4):1055–1085, 2013.
[39] A. Melnikov. Computable topological groups and Pontryagin duality. *Trans. Amer. Math. Soc.*, 370(12):8709–8737, 2018.
[40] A. Melnikov. New degree spectra of Polish spaces. *Sibirsk. Mat. Zh.*, (62:5):1091–1108, 2021.
[41] A. Melnikov and K. M. Ng. Computable structures and operations on the space of continuous functions. *Fund. Math.*, 233(2):101–141, 2016.
[42] A. Melnikov and A. Nies. Computably locally compact totally disconnected groups. Submitted., 2023.
[43] R. Miller, Bjorn Poonen, Hans Schoutens, and Alexandra Shlapentokh. A computable functor from graphs to fields. *J. Symb. Log.*, 83(1):326–348, 2018.
[44] Takakazu Mori, Yoshiki Tsujii, and Mariko Yasugi. Computability structures on metric spaces. In Douglas S. Bridges, Cristian S. Calude, Jeremy Gibbons, Steve Reeves, and Ian H. Witten, editors, *Combinatorics, Complexity, and Logic*, volume 1 of *Discrete Mathematics and Theoretical Computer Science*, pages 351–362. Springer, Singapore, 1997.
[45] Y. Moschovakis. Recursive metric spaces. *Fund. Math.*, 55:215–238, 1964.
[46] A. Nurtazin. Strong and weak constructivizations, and enumerable families. *Algebra i Logika*, 13:311–323, 364, 1974.
[47] S. P. Odintsov and V. L. Selivanov. The arithmetical hierarchy and ideals of enumerated Boolean algebras. *Sibirsk. Mat. Zh.*, 30(6):140–149, 1989.
[48] Arno Pauly. On the topological aspects of the theory of represented spaces. *Computability*, 5(2):159–180, 2016.
[49] M. Pour-El and I. Richards. *Computability in analysis and physics*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1989.
[50] R. Soare. *Recursively enumerable sets and degrees*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987. A study of computable functions and computably generated sets.
[51] Andrea Sorbi, editor. *Complexity, Logic, and Recursion Theory*, volume 187 of *Lecture Notes in Pure and Applied Mathematics*. Marcel Dekker, New York, 1997.
[52] Daniel Turetsky. Coding in the automorphism group of a computably categorical structure. *Journal of Mathematical Logic*, 20(32050016), 2020.

School of Mathematics and Statistics, Victoria University of Wellington, New Zealand
Email address: greenberg@sms.vuw.ac.nz

School of Mathematics and Statistics, Victoria University of Wellington, New Zealand
Email address: alexander.g.melnikov@gmail.com

School of Mathematics and Statistics, Victoria University of Wellington, New Zealand
Email address: isabella.scott@vuw.ac.nz

School of Mathematics and Statistics, Victoria University of Wellington, New Zealand
Email address: dan.turetsky@vuw.ac.nz