LINEAR ORDERS WITH DISTINGUISHED FUNCTION SYMBOL

DOUGLAS CENZER, BARBARA F. CSIMA, AND BAKHADYR KHOUSSAINOV

Abstract. We consider certain linear orders with a function on them, and discuss for which types of functions the resulting structure is or is not computably categorical. Particularly, we consider computable copies of the rationals with a fixed-point free automorphism, and also ω with a non-decreasing function.

May 2, 2008

1. Introduction

We say that computable structures $\mathcal{A}_1$ and $\mathcal{A}_2$ have the same computable isomorphism type if there is a computable isomorphism between them. Existence of an isomorphism between computable structures does not always imply that there is a computable isomorphism between them.

Let $\mathcal{A}$ be a structure. If $\mathcal{B}$ is computable and is isomorphic to $\mathcal{A}$ then $\mathcal{B}$ is called a computable presentation (or copy) of $\mathcal{A}$. The number of computable isomorphism types of $\mathcal{A}$, denoted by $\dim(\mathcal{A})$, is called the computable dimension of $\mathcal{A}$. It is obvious that $\dim(\mathcal{A}) = 1$ if and only if any two computable presentations of $\mathcal{A}$ are computably isomorphic. In case $\dim(\mathcal{A}) = 1$, then we say that $\mathcal{A}$ is computably categorical.

One of the central topics in computable model theory is concerned with the study of computable dimensions of structures and characterizations of computable categoricity. Here we provide several examples. Goncharov proved that for any $n \in \omega \cup \{\omega\}$ there exists a structure of computable dimension n [4]. In [2] Cholak, Goncharov, Khoussainov and Shore gave an example of a computably categorical structure $\mathcal{A}$ such that for each $a \in A$ the structure $(\mathcal{A}, a)$ has computable dimension n, where $n \in \omega$. Goncharov and Remmel proved that a Boolean algebra $\mathcal{A}$ is computably categorical if and only if it has finitely many atoms [7, 3]; similarly a linearly ordered set is computably categorical if and only if the set of successive pairs in the order is finite [6, 3]. Calvert, Cenzer, Harizanov and Morozov [1] show that an equivalence structure is computably categorical if and only if there is a

D. Cenzer was partially supported by National Science Foundation grants DMS 0532644 and 0554841 and 652372.

B. Csima was partially supported by Canadian NSERC Discovery Grant 312501.

B. Khoussainov has partially been supported by Marsden Fund of Royal New Zealand Society.

bound b on the sizes of finite equivalence classes, and there is at most one $t \in \{1, ..., b\} \cup \{\omega\}$ with infinitely many classes of size t. In [5] Khoussainov provided examples of structures of type $(\mathcal{A}, h)$ where h is a function from A to A, of computable dimension n with $n \in \omega$. In [8] Ventsov studied computable dimensions of $(L; \leq, P)$ where $(L; \leq)$ is a l.o. set and P is a unary predicate. This paper is a continuation of the above work with an emphasis to study computable dimensions of linearly ordered sets with distinguished endomorphisms.

In this paper we are interested in structures of the type $\mathcal{A} = (A, <^{\mathcal{A}}, h^{\mathcal{A}})$ where $(A, <^{\mathcal{A}})$ is a linearly ordered set and $h^{\mathcal{A}}$ is a function. We call the structures $\mathcal{A} = (A, <^{\mathcal{A}}, h^{\mathcal{A}})$ described linearly ordered (l.o.) sets with distinguished function symbol. All structures we consider are countable.

A structure $\mathcal{A}$ is computable if its open diagram is a computable set. It is clear that a linearly ordered set $\mathcal{A} = (A, <^{\mathcal{A}}, h^{\mathcal{A}})$ with function $h^{\mathcal{A}}$ is computable if and only if A, $<^{\mathcal{A}}$, and $h^{\mathcal{A}}$ are all computable. For infinite computable structures we may always assume that they have domain ω.

In this paper we concentrate on two types of linearly ordered sets with distinguished function symbol. The first will be the structures of the form $\mathcal{Q} = (\omega, <^{\mathcal{Q}}, h^{\mathcal{Q}})$, where $(\omega, <^{\mathcal{Q}}) \cong \eta$ is the order of rationals and $h^{\mathcal{Q}}$ is a fixed point free automorphism; these structures will be dealt with in the next section. The second will be structures $(\omega; \leq, h)$, where $(\omega, \leq)$ is the standard copy of ω and $h : \omega \to \omega$ is non-decreasing. We note that the successivity relation in the standard copy is decidable. The last section will study these structures. Both sections will investigate computable categoricity.

2. Rationals with distinguished automorphism

Let $\mathcal{Q} = (\omega, <^{\mathcal{Q}}, h^{\mathcal{Q}})$ be a computable structure where $(\omega, <^{\mathcal{Q}})$ has order type η and $h^{\mathcal{Q}}$ is an automorphism with no fixed points. That is, $h^{\mathcal{Q}} : \omega \to \omega$ is bijective, and for all $x, y \in \omega$, $x <^{\mathcal{Q}} y \iff h^{\mathcal{Q}}(x) <^{\mathcal{Q}} h^{\mathcal{Q}}(y)$, and $x \neq h^{\mathcal{Q}}(x)$. From now on in this section, whenever we write $\mathcal{Q} = (\omega, <^{\mathcal{Q}}, h^{\mathcal{Q}})$, we mean such a structure.

For an element $q \in \omega$, consider the sequence $q, h^{\mathcal{Q}}(q), (h^{\mathcal{Q}})^{-1}(q), h^{\mathcal{Q}}(h^{\mathcal{Q}}(q)), (h^{\mathcal{Q}})^{-1}((h^{\mathcal{Q}})^{-1}(q)), ...$ We call the sequence the *orbit of q in $\mathcal{Q}$*. Note that since $h^{\mathcal{Q}}$ is computable and an automorphism, $(h^{\mathcal{Q}})^{-1}$ is also computable, so the orbit of q is computably enumerable. We say that an element $x \in \omega$ is *covered* by the orbit of q if there exist $n, m \in \mathbb{Z}$ such that $(h^{\mathcal{Q}})^n(q) \leq^{\mathcal{Q}} x \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^m(q)$. Let $C_{\mathcal{Q}}$ be the relation $\{(x, y) \mid y \text{ is covered by } x\}$. The following lemma is easy to check.

Lemma 2.1. *The relation $C_{\mathcal{Q}}$ is a c.e. equivalence relation. Moreover for all $x_1, y_1, x_2, y_2 \in \omega$ if $x_1 \leq^{\mathcal{Q}} x_2$ and $(x_1, y_1), (x_2, y_2) \in C_{\mathcal{Q}}$ and $(x_1, x_2) \notin C_{\mathcal{Q}}$ then $y_1 <^{\mathcal{Q}} y_2$.*

Proof. Dovetailing the enumerations of the orbits of each $q \in \omega$, along with comparing each $x \in \omega$ with each member of each orbit under $<^{\mathcal{Q}}$, we see that $C_{\mathcal{Q}}$ is certainly computably enumerable.

It is reflexive since for any $x \in \omega$, $(h^{\mathcal{Q}})^0(x) = x = (h^{\mathcal{Q}})^0(x)$. It is symmetric, since if $(h^{\mathcal{Q}})^n(x) \leq^{\mathcal{Q}} y \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^m(x)$ then $(h^{\mathcal{Q}})^{-m}(y) \leq^{\mathcal{Q}} x \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^{-n}(y)$. It is transitive, since if $(h^{\mathcal{Q}})^n(x) \leq^{\mathcal{Q}} y \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^m(x)$ and $(h^{\mathcal{Q}})^l(y) \leq^{\mathcal{Q}} z \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^k(y)$, then $(h^{\mathcal{Q}})^{n+l}(x) \leq^{\mathcal{Q}} z \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^{k+m}(x)$.

Suppose $x_1, y_1, x_2, y_2 \in \omega$, $x_1 \leq^{\mathcal{Q}} x_2$, $(x_1, y_1), (x_2, y_2) \in C_{\mathcal{Q}}$, and $y_2 \leq^{\mathcal{Q}} y_1$. Then since $(x_1, y_1), (x_2, y_2) \in C_{\mathcal{Q}}$, there exist $n_1, m_1, n_2, m_2 \in \mathbb{Z}$ such that $(h^{\mathcal{Q}})^{n_1}(x_1) \leq^{\mathcal{Q}} y_1 \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^{m_1}(x_1)$ and $(h^{\mathcal{Q}})^{n_2}(x_2) \leq^{\mathcal{Q}} y_2 \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^{m_2}(x_2)$. So since $y_2 \leq^{\mathcal{Q}} y_1$, we have $(h^{\mathcal{Q}})^{n_2}(x_2) \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^{m_1}(x_1)$. Thus since $h^{\mathcal{Q}}$ is an automorphism and $x_1 \leq^{\mathcal{Q}} x_2$, we have $(h^{\mathcal{Q}})^{n_2-m_1}(x_2) \leq^{\mathcal{Q}} x_1 \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^0(x_2)$, so $(x_1, x_2) \in C_{\mathcal{Q}}$. □

Also note that for all $x \in \omega$ we have $(x, h^{\mathcal{Q}}(x)) \in C_{\mathcal{Q}}$. Consider the factor set $\omega/C_{\mathcal{Q}}$. By the lemma above the relation $<^{\mathcal{Q}}$ induces a strict linear order on $\omega/C_{\mathcal{Q}}$. We denote it by $<_{C_{\mathcal{Q}}}$.

The following is also an easy lemma.

Lemma 2.2. *The relation $\leq_{C_{\mathcal{Q}}} = <^{\mathcal{Q}} \cup C_{\mathcal{Q}}$ is a c.e. pre-linear order on ω. That is, $\leq_{C_{\mathcal{Q}}}$ satisfies the following:*

1. *$\leq_{C_{\mathcal{Q}}}$ is computably enumerable;*
2. *$\leq_{C_{\mathcal{Q}}}$ is reflexive;*
3. *$\leq_{C_{\mathcal{Q}}}$ is transitive;*
4. *for all $x, y \in \omega$ at least one of $x \leq_{C_{\mathcal{Q}}} y$ or $y \leq_{C_{\mathcal{Q}}} x$ holds.*

Lemma 2.3. *If the equivalence relation $C_{\mathcal{Q}}$ of the computable structure $\mathcal{Q} = (\omega, <^{\mathcal{Q}}, h^{\mathcal{Q}})$ has a finite index then $\mathcal{Q}$ is computably categorical.*

Proof. Note that the hypothesis of the lemma is equivalent to saying that the linearly ordered set $(\omega/C_{\mathcal{Q}}, <_{C_{\mathcal{Q}}})$ is finite, say of size k. Let $\mathcal{A} = (\omega, <^{\mathcal{A}}, h^{\mathcal{A}})$ be isomorphic to $\mathcal{Q}$. Let $q_1 <^{\mathcal{Q}} ... <^{\mathcal{Q}} q_k$ be representatives of the k distinct $C_{\mathcal{Q}}$-equivalence classes in $\mathcal{Q}$. Let $a_1 <^{\mathcal{A}} ... <^{\mathcal{A}} a_k$ be their images under some isomorphism $\mathcal{A} \cong \mathcal{Q}$. We use this finite information to build a computable isomorphism $f : \mathcal{Q} \to \mathcal{A}$ using a standard back-and-forth argument. Recall that both structures have domain ω, so as we build f by stages we may speak of the least number not yet in the domain/range of f. At each stage s we will ensure that f_s is a partial isomorphism on its domain. That is, if $x, z \in \mathrm{dom}(f_s)$ and $n, m \in \mathbb{Z}$, we will have $(h^{\mathcal{Q}})^n(x) \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^m(z) \iff (h^{\mathcal{A}})^n(f_s(x)) \leq^{\mathcal{A}} (h^{\mathcal{A}})^m(f_s(z))$. Similarly for $x, z \in \mathrm{rng}(f_s)$ we will have $(h^{\mathcal{A}})^n(x) \leq^{\mathcal{A}} (h^{\mathcal{A}})^m(z) \iff (h^{\mathcal{Q}})^n(f_s^{-1}(x)) \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^m(f_s^{-1}(z))$.

Stage 0: Let $f_0(q_i) = a_i$ for $i = 1, ..., k$.

Stage $s + 1 = 2l + 1$: Let x be least such that $x \notin \mathrm{dom}(f_s)$. Enumerate the orbits of $q_1, ..., q_k$ until we find i such that x is covered by q_i. Since h has no fixed points, either $h^{\mathcal{Q}}(q_i) <^{\mathcal{Q}} q_i$ or $q_i <^{\mathcal{Q}} h^{\mathcal{Q}}(q_i)$. Assume w.l.o.g. that $q_i <^{\mathcal{Q}} h^{\mathcal{Q}}(q_i)$, so that $h^{\mathcal{Q}}$ is strictly increasing on $[q_i]$. Let $\{x_1, x_2, ..., x_d\} = \mathrm{dom}(f_s) \cap [q_i]$. For each x_j we can compute $n_j \in \mathbb{Z}$ such that $(h^{\mathcal{Q}})^{n_j}(x_j) \leq^{\mathcal{Q}} x <^{\mathcal{Q}} (h^{\mathcal{Q}})^{n_j+1}(x_j)$.

If $x = (h^{\mathcal{Q}})^{n_j}(x_j)$ for some j, then define $f_{s+1}(x) = (h^{\mathcal{A}})^{n_j}(f_s(x_j))$. Note that if $x_j, x_k \in \mathrm{dom}(f_s)$ and $(h^{\mathcal{Q}})^{n_j}(x_j) = (h^{\mathcal{Q}})^{n_k}(x_k)$, then by induction

hypothesis $(h^{\mathcal{A}})^{n_j}(f_s(x_j)) = (h^{\mathcal{A}})^{n_k}(f_s(x_k))$, so that $f_{s+1}(x)$ is well defined. Also note that $(h^{\mathcal{A}})^{n_j}(f_s(x_j)) \notin \mathrm{rng}(f_s)$ since if there were $p \in \mathrm{dom}(f_s)$ such that $f_s(p) = (h^{\mathcal{A}})^{n_j}(f_s(x_j))$ then by induction hypothesis $p = (h^{\mathcal{Q}})^{n_j}(x_j)$, a contradiction since $(h^{\mathcal{Q}})^{n_j}(x_j) = x \notin \mathrm{dom}(f_s)$. Thus f_{s+1} defined in this way is injective.

Otherwise, we have $(h^{\mathcal{Q}})^{n_j}(x_j) <^{\mathcal{Q}} x <^{\mathcal{Q}} (h^{\mathcal{Q}})^{n_j+1}(x_j)$ for all $1 \leq j \leq d$. Let $(h^{\mathcal{Q}})^{n_k}(x_k)$ be $\mathcal{Q}$-maximal such that $(h^{\mathcal{Q}})^{n_k}(x_k) <^{\mathcal{Q}} x$. Let $(h^{\mathcal{Q}})^{n_l}(x_l)$ be $\mathcal{Q}$-minimal such that $x <^{\mathcal{Q}} (h^{\mathcal{Q}})^{n_l}(x_l)$. Then since $(h^{\mathcal{Q}})^{n_k}(x_k) <^{\mathcal{Q}} (h^{\mathcal{Q}})^{n_l}(x_l)$, by induction hypothesis we have $(h^{\mathcal{A}})^{n_k}(f_s(x_k)) <^{\mathcal{A}} (h^{\mathcal{A}})^{n_l}(f_s(x_l))$. Choose the least y such that $(h^{\mathcal{A}})^{n_k}(f_s(x_k)) <^{\mathcal{A}} y <^{\mathcal{A}} (h^{\mathcal{A}})^{n_l}(f_s(x_l))$, and define $f_{s+1}(x) = y$. This clearly makes f_{s+1} well defined and injective on its domain. It remains to check that f_{s+1} still satisfies the induction hypothesis. First note that for any x_j and any $n \in \mathbb{Z}$, if $(h^{\mathcal{Q}})^n(x_j) <^{\mathcal{Q}} x$ then by definition of n_j and k, $(h^{\mathcal{Q}})^n(x_j) \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^{n_j}(x_j) \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^{n_k}(x_k) <^{\mathcal{Q}} x$. By induction hypothesis, $(h^{\mathcal{A}})^n(f_s(x_j)) \leq^{\mathcal{A}} (h^{\mathcal{A}})^{n_k}(f_s(x_k))$, and by definition of $f_{s+1}(x)$, $(h^{\mathcal{A}})^{n_k}(f_s(x_k)) <^{\mathcal{A}} f_{s+1}(x)$. Thus for any $x_j \in \mathrm{dom} f_s \cap [q_i]$ and any $n, m \in \mathbb{Z}$ we have:

$$\begin{aligned}(h^{\mathcal{Q}})^n(x_j) \leq^{\mathcal{Q}} &(h^{\mathcal{Q}})^m(x)\\ &\Rightarrow (h^{\mathcal{Q}})^{n-m}(x_j) \leq^{\mathcal{Q}} x\\ &\Rightarrow (h^{\mathcal{A}})^{n-m}(f_{s+1}(x_j)) \leq^{\mathcal{A}} f_{s+1}(x) \text{ (by the above argument)}\\ &\Rightarrow (h^{\mathcal{A}})^n(f_{s+1}(x_j) \leq^{\mathcal{A}} (h^{\mathcal{A}})^m(f_{s+1}(x))\end{aligned}$$

A similar argument shows that

$$(h^{\mathcal{Q}})^n(x) \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^m(x_j) \Rightarrow (h^{\mathcal{A}})^n(f_{s+1}(x)) \leq^{\mathcal{A}} (h^{\mathcal{A}})^m(f_{s+1}(x_j)).$$

Again similarly we have:

$$\begin{aligned}(h^{\mathcal{A}})^n(f_{s+1}(x_j)) \leq^{\mathcal{A}} &(h^{\mathcal{A}})^m(f_{s+1}(x))\\ &\Rightarrow (h^{\mathcal{A}})^{n-m}(f_{s+1}(x_j)) \leq^{\mathcal{A}} f_{s+1}(x)\\ &\Rightarrow (h^{\mathcal{Q}})^{n-m}(x_j) \leq^{\mathcal{Q}} f_{s+1}(x)\\ &\Rightarrow (h^{\mathcal{Q}})^n(x_j) \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^m(x),\end{aligned}$$

where the second implication follows since $x < (h^{\mathcal{Q}})^{n-m}(x_j)$ implies $f_{s+1}(x) \leq^{\mathcal{A}} (h^{\mathcal{A}})^{n-m}(f_{s+1}(x_j))$.

Now for $\hat{x} \in \mathrm{dom} f_s$ such that $\hat{x} \notin [q_i]$, then $\hat{x} \in [q_{\hat{i}}]$ for some $\hat{i} \neq i$. But then as we have seen, we have defined $f_s(\hat{x}) \in [a_{\hat{i}}]$. Thus

$$\begin{aligned}
&(h^{\mathcal{Q}})^n(\hat{x}) \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^m(x_j) \\
&\iff q_{\hat{i}} <^{\mathcal{Q}} q_i \\
&\iff a_{\hat{i}} \leq^{\mathcal{A}} a_i \\
&\iff (h^{\mathcal{A}})^n(f_s(\hat{x})) \leq^{\mathcal{A}} (h^{\mathcal{A}})^m(f_{s+1}(x_j))
\end{aligned}$$

Similarly,$(h^{\mathcal{Q}})^n(x_j) \leq^{\mathcal{Q}} (h^{\mathcal{Q}})^m(\hat{x}) \iff (h^{\mathcal{A}})^n(f_{s+1}(x_j)) \leq^{\mathcal{A}} (h^{\mathcal{A}})^m(f_s(\hat{x}))$.

Thus we have maintained that for any $x, z \in \mathrm{dom} f_{s+1}$, and any $m, n \in \mathbb{Z}$, $(h^{\mathcal{Q}})^n(x) \leq (h^{\mathcal{Q}})^m(z) \iff (h^{\mathcal{A}})^n(f_{s+1}(x)) \leq^A (h^{\mathcal{A}})^m(f_{s+1}(z))$.

Stage $s+1 = 2l+2$*:* As above with $\mathrm{dom}(f_s)$ replaced by $\mathrm{rng}(f_s)$, $q_1, ..., q_k$ replaced by $a_1, ..., a_k$, and f_s replaced by f_s^{-1}.

Let $f = \cup_s f_s$. Then f is total and onto since $f(x)$ and $f^{-1}(x)$ are defined by stage $2x+2$. The fact that f is an isomorphism follows from the fact that at each stage it was a partial isomorphism. □

Now our goal is to show the converse of the lemma above.

Definition 2.4. A *computably enumerable pre-linear order* is a structure of the form (ω, R), where R satisfies the following properties:

(1) R is computably enumerable;
(2) R is reflexive;
(3) R is transitive;
(4) for all $x, y \in \omega$ either $(x, y) \in R$ or $(y, x) \in R$.

By Lemma 2.2, if $\mathcal{Q} = (\omega, <^{\mathcal{Q}}, h^{\mathcal{Q}})$ is a computable structure of the type we have been discussing, then the structure $(\omega, \leq_{C_{\mathcal{Q}}})$ is a computably enumerable pre-linear order.

For a c.e. pre-linear order (ω, R), define $\equiv_R$ to be $\{(x, y) \mid (x, y) \in R \text{ and } (y, x) \in R\}$. Clearly, $\equiv_R$ is a c.e. equivalence relation. Moreover, R induces a linear order $(\omega/\equiv_R, <_R)$, where $<_R$ is induced by R on the equivalence classes of the relation $\equiv_R$.

We say that two pre-linearly ordered c.e. structures (ω, R) and (ω, S) are computably isomorphic if there exists a computable function $f : \omega \to \omega$ that induces an isomorphism between the linearly ordered sets $(\omega/\equiv_R, <_R)$ and $(\omega/\equiv_S, <_S)$. Note that f need not be a bijective function on ω. We also note that each pre-linearly ordered set (ω, R) is computably isomorphic to (ω, R') such that each equivalence class $\equiv_{R'}$ of R' is an infinite set.

A c.e. presentation of a linearly ordered set $\mathcal{L} = (L, <^{\mathcal{L}})$ is a pre-linearly ordered c.e. structure (ω, R) such that $\mathcal{L}$ is isomorphic to $(\omega/\equiv_R, <_R)$.

Theorem 2.5. *A linearly ordered set $\mathcal{L}$ is finite if and only if all c.e. presentations of $\mathcal{L}$ are computably isomorphic.*

Proof. Suppose $\mathcal{L} = (L, <^{\mathcal{L}})$ is finite, of size k, and that (ω, R) and (ω, S) are c.e. presentations of $\mathcal{L}$. Let $r_1, ..., r_k, s_1, ..., s_k \in \omega$ be such that $[r_1] <_R ... <_R$

$[r_k]$ and $[s_1] <_S ... <_S [s_k]$. We define a computable function $f : \omega \to \omega$ as follows. For each $x \in \omega$ there exists some $1 \leq i \leq k$ such that $x \in [r_i]$. Since R is c.e. and $(\omega/\equiv_R, <_R)$ is finite, we can compute i. Set $f(x) = s_i$. Certainly f induces an isomorphism $(\omega/\equiv_R, <_R) \cong (\omega/\equiv_S, <_S)$.

Now suppose $\mathcal{L}$ is infinite and assume by the remark above that each equivalence class of R is also infinite. We will use a priority argument to show that there exist two c.e. presentations of $\mathcal{L}$ that are not computably isomorphic. Let (ω, R) be a c.e. presentation of $\mathcal{L}$. We will build another c.e. presentation of $\mathcal{L}$, (ω, S), that is not computably isomorphic to (ω, R). The strategy to avoid a particular possible computable isomorphism will be to choose elements that appear to be in different R-equivalence classes, and place them into different S-equivalence classes. If the isomorphism says they are different, we will make S collapse them together. The R order might then also collapse them. But after finitely many tries, the R order can no longer follow the collapse, since the order must have infinitely many equivalence classes. With priority, the lower priority requirements should try to act elsewhere, to ensure that the order we build is a c.e. presentation of $\mathcal{L}$.

We now give the formal construction of (ω, S). We assume that (ω, R) is revealed to us stage by stage in a way such that at each stage s, (s, R_s) is a finite pre-linear order, $R_{s+1} \supset R_s$, and $R = \cup_{s\in\omega} R_s$. We will enumerate S stage by stage, to ensure it is computably enumerable. We also build a map $f : \omega \to \omega$ that induces an isomorphism $(\omega/\equiv_S, <_S) \to (\omega/\equiv_R, <_R)$ using stage by stage approximations. At each stage s, we will have $\mathrm{dom} f_s$ an initial segment of ω, and have $\mathrm{dom} f_{s+1} \supset \mathrm{dom} f_s$. We will have $S_s \subseteq (\mathrm{dom} f_s)^2$ be reflexive, transitive, and for all $x, y \in \mathrm{dom} f_s$ have either $(x, y) \in S_s$ or $(y, x) \in S_s$, so that S will be a c.e. pre-linear order. At each stage s, we will define f_s such that f_s induces an isomorphism $(\mathrm{dom} f_s/\equiv_{S_s}, <_{S_s}) \cong (\mathrm{dom} f_s/\equiv_{R_s}, <_{R_s})$. We will ensure that for each $y \in \omega$, $\lim_s f_s(y)$ exists, so that f will induce an isomorphism $(\omega/\equiv_S, <_S) \cong (\omega/\equiv_R, <_R)$. We will also meet for each $e \in \omega$ the requirement:

$$Q_e : \varphi_e \text{ does not induce an isomorphism } (\omega/\equiv_R, <_R) \cong (\omega/\equiv_S, <_S)$$

To meet requirement Q_e, we will at each stage $s \geq e$ have defined $x^e_{0,s} \leq_{S_s} x^e_{1,s}$. The goal will be that for each $e \in \omega$ and $i \in \{0, 1\}$, $x^e_i = \lim_s x^e_{i,s}$ exist, and if $\varphi_e(x^i_e) \downarrow$ for $i = 0, 1$ then if $\varphi_e(x^e_0) \equiv_R \varphi_e(x^e_1)$ then $x^e_0 <_S x^e_1$, and if $\varphi_e(x^e_0) <_R \varphi_e(x^e_1)$ then $x^e_0 \equiv_S x^e_1$, so that φ_e does not induce an isomorphism $(\omega/\equiv_S, \leq_S) \cong (\omega/\equiv_R, \leq_R)$.

Since we are building S to be a pre-linear order, we will often say "insert z into S between x and y" as shorthand for "for all w such that $w \leq_{S_s} x$, enumerate $(w, z) \in S_{s+1}$, and for all w such that $w \geq_{S_s} y$, enumerate $(z, w) \in S_{s+1}$." We will also use "$x \in R_s$" as shorthand for $(x, x) \in R_s$.

Stage 0*:* Let $x^0_0 = 0$ and $x^0_1 = 1$. Speed up the enumeration of R so that in R_0 there exist $u <_{R_0} v$. Enumerate $(0,1) \in S_0$, and define $f_0(0) = u$ and $f_0(1) = v$.

Stage $s+1$*:*

Begin with $\tilde{f} = f_s$. As we go through stage $s+1$ we will make (finitely many) changes to $\tilde{f}$. We will let f_{s+1} be the final version of $\tilde{f}$ at the end of the stage.

Step 1: Begin by letting $\tilde{s} = s$. Choose the least $e \in \omega$ such that (1) or (2) hold.

$$(\exists l < e)(\exists i, j \in \{0,1\})[\tilde{f}(x^e_{i,s}) \equiv_{R_{\tilde{s}}} \tilde{f}(x^l_{j,s})] \tag{1}$$

$$\tilde{f}(x^e_{0,s}) \equiv_{R_{\tilde{s}}} \tilde{f}(x^e_{1,s}) \wedge \neg(\varphi_{e,\tilde{s}}(x^e_{0,s}) \downarrow <_{R_{\tilde{s}}} \varphi_{e,\tilde{s}}(x^e_{i,s}) \downarrow) \tag{2}$$

For $n < e$, set $x^n_{i,s+1} = x^n_{i,s}$. For each $e \leq n \leq s$, in turn, do as follows. Speeding up the enumeration of R to a stage $s' \geq \tilde{s}$ if necessary, find the least $\langle u, v\rangle \in \omega$ such that $u <_{R_{s'}} v$ and there is no $x^l_{j,s+1}$ with $l < e$ and $u \leq_{R_{s'}} \tilde{f}(x^l_{j,s+1}) \leq_{R_{s'}} v$. If $u \in \mathrm{rng}\tilde{f}$, set $x^n_{0,s+1} = \tilde{f}^{-1}(u)$, otherwise choose the least $w \notin \mathrm{dom}\tilde{f}$, set $x^n_{0,s+1} = w$, enumerate $(a,w) \in S_{s+1}$ for all $a \in \mathrm{dom}\tilde{f}$ such that $\tilde{f}(a) \leq_{R_{s'}} v$, enumerate $(w,b) \in S_{s+1}$ for all $b \in \mathrm{dom}\tilde{f}$ such that $v \leq_{R_{s'}} \tilde{f}(v)$, and define $\tilde{f}(w) = u$. Similarly, if $v \in \mathrm{rng}\tilde{f}$, set $x^n_{1,s+1} = \tilde{f}^{-1}(v)$, otherwise choose the least $w \notin \tilde{f}$, set $x^n_{1,s+1} = w$, enumerate $(a,w) \in S_{s+1}$ for all $a \in \mathrm{dom}\tilde{f}$ such that $\tilde{f}(a) \leq_{R_{s'}} u$, enumerate $(w,b) \in S_{s+1}$ for all $b \in \mathrm{dom}\tilde{f}$ such that $u \leq_{R_{s'}} \tilde{f}(b)$, and define $\tilde{f}(w) = v$. (Here we are using the assumption that each equivalence class is infinite.) If in speeding up the enumeration of R we caused $\tilde{f}(x^l_{i,s+1}) \equiv_{R_{s'}} \tilde{f}(x^{l'}_{i',s+1})$ for some $l, l' < n$, then go back to the beginning of step 1 and repeat with $\tilde{s} = s'$. This process must halt since $\mathcal{L}$ is infinite. In step 2, when we refer to R_{s+1}, we really mean the final $R_{s'}$ at the end of step 1.

Step 2: For each $u \in R_{s+1} - \mathrm{rng}\tilde{f}$, in turn, compute $y_0, y_1 \in \mathrm{rng}\tilde{f}$ such that $y_0 <_{R_{s+1}} u <_{R_{s+1}} y_1$ and there is no $z \in \mathrm{rng}\tilde{f}$ with $y_0 <_{R_{s+1}} z <_{R_{s+1}} y_1$. Choose the least $w \notin \mathrm{dom}\tilde{f}$, insert w into S between $\tilde{f}^{-1}(y_0)$ and $\tilde{f}^{-1}(y_1)$, and define $\tilde{f}(w) = u$.

Step 3: Let e be least such that $\varphi_e(x^e_{0,s+1}) \downarrow <_{R_{s+1}} \varphi_e(x^e_{1,s+1}) \downarrow$. Enumerate $(y, x^e_{0,s+1}) \in S_{s+1}$ and re-define $\tilde{f}(y) = \tilde{f}(x^e_{0,s+1})$ for all $x^e_{0,s+1} \leq_{S_{s+1}} y \leq_{S_{s+1}} x^e_{1,s+1}$. For each $u \in R_{s+1} - \mathrm{rng}\tilde{f}$, in turn, compute $y_0, y_1 \in \mathrm{rng}\tilde{f}$ such that $y_0 <_{R_{s+1}} u <_{R_{s+1}} y_1$ and there is no $z \in \mathrm{rng}\tilde{f}$ with $y_0 <_{R_{s+1}} u <_{R_{s+1}} y_1$. Choose the least $w \notin \mathrm{dom}\tilde{f}$, insert w into S between $\tilde{f}^{-1}(y_0)$ and $\tilde{f}^{-1}(y_1)$, and define $\tilde{f}(w) = u$.

Let $f_{s+1} = \tilde{f}$.

Lemma 2.6. *For each $e \in \omega, i \in \{0,1\}$, $x_i^e = \lim_s x_{i,s}^e$ exists, and $f(x_i^e) = \lim_s f_s(x_i^e)$ exists.*

Proof. By induction. Suppose the lemma holds for all $l < e$. Let s_0 be the stage by which $x_{j,s}^l = x_j^l$ and $f_s(x_j^l) = f(x_j^l)$ for all $s \geq s_0$ and $l < e$. Let $\langle u, v\rangle$ be least such that $u <_R v$ and there is no x_j^l with $l < n$ and $u \leq_R f(x_j^l) <_R v$. Such a pair must exist since $\mathcal{L}$ is infinite. Let s_1 be the stage by which $(u, v) \in R_{s_1}$. If $x_{i,s}^e$ were re-defined at any stage $s_2 > s_1$, then they would be set such that $f_{s_2}(x_{0,s_2}^e) = u$ and $f_{s_2}(x_{1,s_2}^e) = v$. Then by induction hypothesis we would never re-define $x_{i,s}^e$. If at some stage $s_3 > s_2$ we found that $\varphi_{e,s_3}(x_{0,s_2}^e) \downarrow <_{R_{s_3}} \varphi_{e,s_3}(x_{0,s_2}^e) \downarrow$ then we would have defined $f_{s_3}(x_1^e) = f_{s_3}(x_1^e)$, and those would be the final values of x_i^e. □

Lemma 2.7. *For each $x \in \omega$, $f(x) = \lim_s f_s(x)$ exists.*

Proof. The only time $f_s(x)$ is re-defined is in step 3 of the construction. If that happens, we also set $x \equiv_{S_s} x_{i,s}^e$ for some $e \in \omega$, $i \in \{0,1\}$. Then since f_s is induces an isomorphism at every stage of the construction, and since $\lim_s f_s(x_{i,s}^e)$ exists, we must have $f(x) = f(x_i^e)$. □

This completes the proof of Theorem 2.5. □

Lemma 2.8. *Let (ω, R) be a c.e. pre-linearly ordered set. There exists a computable structure $\mathcal{Q} = (\omega, <^{\mathcal{Q}}, h^{\mathcal{Q}})$ such that $(\omega, \leq_{C_{\mathcal{Q}}})$ is computably isomorphic to (ω, R).*

Proof. Note that if we take a computable copy of the rationals with addition, then taking $h(x) = x + 1$ covers all elements. We name this structure $\mathbb{Q}_{+1}$. Since (ω, R) is a c.e. pre-linearly ordered set, we may assume it is revealed to us in such a way that at stage s, for any $n, m \leq s$, either $(n, m) \in R_s$ or $(m, n) \in R_s$. Recall that since R is a c.e. pre-linearly ordered set, if at stage s we believe that $n <_{R_s} m$, then we might find out at a later stage t that $n \equiv_{R_t} m$. We will construct a computable structure $\mathcal{Q} = (\omega, <^{\mathcal{Q}}, h^{\mathcal{Q}})$ by stages. At each stage s, we will decide where a bunch of numbers will sit in the $<^{\mathcal{Q}}$ ordering, and by the next stage at the latest, we will define $h^{\mathcal{Q}}$ and $(h^{\mathcal{Q}})^{-1}$ on them. When we enumerate numbers, we will give them labels. As we proceed through the construction, the labels of the numbers will change (though of course their position in the ordering and the values of $h^{\mathcal{Q}}$ on them will not). For each equivalence class in R, we want to build a copy of $\mathbb{Q}_{+1}$. As we find out that members we thought were not R-equivalent turn out to be R-equivalent after all, we must link the copies we are building. We will define a computable function $f : \omega \to \omega$ stage by stage.

If $n \leq s$ is such that there is no $m < n$ with $m \leq_{R_s} n$, then we ensure that n has an "active label group" at stage s. An active label group will be a set of numbers in ω with labels $\langle n, \frac{k}{2^s}\rangle$ with the following properties. There will exist $m_{n,s} \leq M_{n,s} \in \mathbb{Z}$ such that for all $m_{n_s} \leq k < M_{n,s} + 2^s$, there are numbers with label $\langle n, \frac{k}{2^s}\rangle$ such that $\langle n, \frac{l}{2^s}\rangle <^{\mathcal{Q}} \langle n, \frac{k}{2^s}\rangle \iff l < k$, and $h^{\mathcal{Q}}(\langle n, \frac{k}{2^s}\rangle) = \langle n, \frac{k}{2^s} + 1\rangle$ for $m_n \leq k < M_n$.

Stage 0: Give 0 the label $\langle 0,0\rangle$. Set $f(0)=0$.

Stage s+1:

Step 1: If there exist $n,p\leq s$ such that $n<_{R_s}p$ but $n\equiv_{R_{s+1}}p$, then we must join the n and p label groups. We set $h(\langle n,\frac{M_{n,s}+k}{2^s}\rangle)=\langle p,\frac{m_{p,s}+k}{2^s}\rangle$. If $n<p$ then we give all the numbers with p-labels new n-labels by setting $\langle p,\frac{m_{p,s}+k}{2^s}\rangle=\langle n,\frac{M_{p,s}+k}{2^s}+1\rangle$. If $p<n$ then we give all the numbers with n-labels new p-labels by setting $\langle n,\frac{M_{n,s}+k}{2^s}\rangle=\langle p,\frac{m_{p,s}+k}{2^s}-1\rangle$.

Step 2: If $s+1\equiv_{R_{s+1}}n$ for some $n\leq s$, then assume n is the least such. In that case there is an active n-label group. Set $f(s+1)=\langle n,0\rangle$. If $s+1\not\equiv_{R_{s+1}}n$ for any $n\leq s$, then we introduce a new $s+1$-label group. We take the next 2^s many numbers in ω that have not yet been used, and give them labels $\langle s+1,\frac{k}{2^s}\rangle$ where $0\leq k<2^s$. We declare $\langle s+1,i\rangle<^{\mathcal{Q}}\langle s+1,j\rangle\iff i<j$, $\langle n,i\rangle<^{\mathcal{Q}}\langle s+1,j\rangle$ if $n<_{R_{s+1}}s+1$, and $\langle s+1,i\rangle<^{\mathcal{Q}}\langle n,j\rangle$ if $s+1<_{R_{s+1}}n$. We let $f(s+1)=\langle s+1,0\rangle$.

Step 3: We extend each active n-label group. That is, for each active n-label group, we let $m_{n,s+1}=2m_{n,s}-1$ and $M_{n,s+1}=2M_{n,s}+1$. For those k with $m_{n,s+1}\leq k<M_{n,s+1}+2^{s+1}$ and no number with label $\langle n,\frac{k}{2^{s+1}}\rangle$, we insert new numbers with labels $\langle n,\frac{k}{2^{s+1}}\rangle$. We declare $\langle n,i\rangle<^{\mathcal{Q}}\langle n,j\rangle\iff i<j$. For the new numbers with labels $\langle n,\frac{k}{2^{s+1}}\rangle$ and $m_{n,s+1}\leq k<M_{n,s+1}$, we set $h(\langle n,i\rangle)=\langle n,i+1\rangle$.

In each R-equivalence class, there is a least number. Suppose x is the least number in an R-equivalence class. Then at stage x of the construction, we create (either via step 1 or step 2) an active x-label group. Since x is least in its R-equivalence class, its label group will never be deactivated by step 1 at any later stage. Through the Step 3s of successive stages, there will be a copy of $\mathbb{Q}_{+1}$ built with x-labels.

Suppose $x\leq_R y$. Then at step 2 of stage $s=\max\{x,y\}$ of the construction, we ensure that $f(x)\leq^{\mathcal{Q}}f(y)$. If at that stage $x\equiv_{R_s}y$, then we in fact ensured $f(x)\equiv_{C_{\mathcal{Q}}}f(y)$. If at that stage $x<_{R_s}y$, then we had $f(x)$ and $f(y)$ in distinct label groups. These label groups could only be linked by step 1 at a later stage if we found that $x\equiv_R y$. If we never saw $x\equiv_R y$, then the label groups would never be linked, and so they would give rise to distinct copies of $\mathbb{Q}_{+1}$. Thus f induces a computable isomorphism from (ω,R) to $(\omega,\leq_{C_{\mathcal{Q}}})$.

□

Lemma 2.9. *Assume $\mathcal{Q}=(\omega,<^{\mathcal{Q}},h^{\mathcal{Q}})$ is such that $(\omega/C_{\mathcal{Q}},<_{C_{\mathcal{Q}}})$ is not a finite linear order. Then $\mathcal{Q}$ is not computably categorical.*

Proof. Consider $(\omega,\leq_{C_{\mathcal{Q}}})$. It is a c.e. pre-linear order. Let (ω,R) be a c.e. pre-linear order isomorphic but not computably isomorphic to $(\omega,\leq_{C_{\mathcal{Q}}})$. Such (ω,R) exists by Theorem 2.5. By Lemma 2.8, there exists $\mathcal{A}=(\omega,<^{\mathcal{A}},h^{\mathcal{A}})$ such that (ω,R) and $(\omega,C_{\mathcal{A}})$ are computably isomorphic. It is clear that $\mathcal{Q}$ and $\mathcal{A}$ are isomorphic. If they were computably isomorphic then the c.e. pre-linear orders $(\omega,\leq_{C_{\mathcal{Q}}})$ and $(\omega,\leq_{C_{\mathcal{A}}})$ would also be computably

isomorphic. But this would imply that $(\omega, \leq_{C_\mathcal{Q}})$, and (ω, R) are computably isomorphic, a contradiction. □

Theorem 2.10. *Let $\mathcal{Q} = (\omega, <^\mathcal{Q}, h^\mathcal{Q})$ be a computable structure such that $(\omega, <^\mathcal{Q}) \cong \eta$ and $h^\mathcal{Q}$ is an automorphism without fixed points. Then $\mathcal{Q}$ is computably categorical if and only if $(\omega, \leq_{C_\mathcal{Q}})$ is a finite linear order.*

Proof. By Lemma 2.3 and Lemma 2.9. □

3. Natural numbers with distinguished endomorphisms

Now we consider $(\omega, \leq, h)$, where h is a non-decreasing function.

We first recall that $(\omega, \leq)$ is not computably categorical. The main difference between $(\omega, \leq)$ and other computable copies that are isomorphic to it is that in the standard copy the successivity relation is decidable, but this is not true for arbitrary copies. Recall that to show $(\omega, \leq)$ is not computably categorical, we build stage by stage a computable copy where we occasionally insert an extra point to kill an isomorphism. So long as we don't insert infinitely many points below a fixed one, we end up building a copy of ω.

Now consider the case of $(\omega, \leq, h)$, where h is a monotonic function. Notice that if h is just the identity, then we are in the case of $(\omega, \leq)$, and $(\omega, \leq, h)$ is not computably categorical. On the other hand, if for all $x \in \omega$, $h(x) = x + 1$, and $\mathcal{A} \cong (\omega, \leq)$ then mapping $f(0) = 0^\mathcal{A}$ and $f(x+1) = h^\mathcal{A}(f(x))$ gives a computable isomorphism, so in this case $(\omega, \leq, h)$ is computably categorical.

In the second example, every number in ω was linked to 0 via h. We define the *trace of x in $(\omega, \leq, h)$* to be the set $\{h^n(x) \mid n \in \omega\}$. The next theorem shows that if the trace of any member of $(\omega, \leq, h)$ is infinite, then $(\omega, \leq, h)$ is computably categorical.

Theorem 3.1. *If there exists $x \in \omega$ such that $(\forall n \ \in \omega)[h^{n+1}(x) > h^n(x)]$, then $(\omega, \leq, h)$ is computably categorical.*

Proof. Suppose $\mathcal{A} \cong (\omega, \leq, h)$, and suppose $f : (\omega, \leq, h) \to \mathcal{A}$ is an isomorphism. We show how to compute f given $f(x)$, where x is such that $(\forall n \ \in \omega)[h^{n+1}(x) > h^n(x)]$. To compute $f(y)$ for arbitrary $y \in \omega$, first compute n such that $h^n(x) > y$. We then compute $h^n(f(x))$. Then search until we find $z_0 <^\mathcal{A} z_1 <^\mathcal{A} ... <^\mathcal{A} z_y <^\mathcal{A} ... <^\mathcal{A} z_{h^n(x)} = h^n(f(x))$. Since $\mathcal{A} \cong (\omega, \leq, h)$, we will know that our approximation to $\mathcal{A}$ is correct up to $h^n(x)$, so we know that $f(y) = z_y$. □

Consider $(\omega, \leq, h)$. If the trace of f on 0 is not infinite, then its trace defines a "clump" that begins with 0 and ends with $h^n(0)$ where $h^n(0) = h^{n+1}(0)$. More precisely, we define the clumps of $(\omega, \leq, h)$ as follows. Let $\mathcal{C}_0 = \{x \mid (\exists n)[x \leq h^n(0)]\}$. If $\max\{\mathcal{C}_i\} < \infty$, we let $\mathcal{C}_{i+1} = \{x \mid x > \max\{\mathcal{C}_i\} \& (\exists n)[x \leq h^n(\max\{\mathcal{C}_i\} + 1)]\}$.

We now describe computable categoricity of $(\omega, \leq, h)$ in terms of its clumps. Theorem 3.1 shows that if $(\omega, \leq, h)$ has an infinite clump, then

it is computably categorical. In the next theorem, we see that if the size of the clumps is bounded by a constant, then $(\omega, \leq, h)$ is not computably categorical. (Actually, the next theorem shows more, since bounded trace does not imply bounded clumps, though the converse is obviously true). We thank Jim Geelen for pointing out that the combinatorial fact we need comes from Higman's Lemma.

Theorem 3.2. *If there exists $b \in \omega$ such that for all $x \in \omega$, $h^{b+1}(x) = h^b(x)$, then $(\omega, \leq, h)$ is not computably categorical.*

Proof. Let $\mathcal{A} = (\omega, \leq, h)$ be given with such a bound b on the trace, and let $\mathcal{C}_0, \mathcal{C}_1, \ldots$ be as defined above. We first prove the following claim.

Claim 3.3. *There exists some $N \in \omega$ such that $(\forall i > N)(\exists j > i)[\mathcal{C}_i \hookrightarrow \mathcal{C}_j]$.*

Proof. To any finite clump $\mathcal{C}$, we can associate a finite tree $\mathcal{T}$, with labels the members of $\mathcal{C}$, as follows. Since $\mathcal{C}$ is a finite clump, it has a unique fixed point with respect to h. We let this fixed point label the root of the tree $\mathcal{T}$, and call this level 0 of $\mathcal{T}$. In general, if x is the label of a node at level n of $\mathcal{T}$, then we let the node with label x have children with labels y, $y+1$,...,$y+m$, listed in increasing order from left to right, where $h^{-1}(x) = \{y, ..., y+m\}$ (we know that $h^{-1}(x)$ consists of consecutive numbers since h is non-decreasing). Also since h is non-decreasing, if $x < z$ then all children of x are less than all children of z. Furthermore, all labels on level $n+1$ of $\mathcal{T}$ are less than all labels on level n. It is now easy to see that if we define an embedding of finite trees $\mathcal{T} \hookrightarrow \mathcal{T}'$ to be a map that sends the root to the root, maps children of a node to children of the image of the node, and preserves left-to-right ordering of nodes, then an embedding of finite clumps corresponds exactly to an embedding of the corresponding finite trees, and conversely. Our sequence of clumps $\mathcal{C}_0, \mathcal{C}_1, ...$ can thus be viewed as a sequence of trees $\mathcal{T}_0, \mathcal{T}_1, ...$, where the bound b on the trace of the function h translates into a bound b on the height of each tree in the sequence. The claim now follows from b applications of Higman's Lemma. □

Hence we may assume without loss of generality that *every* clump in $\mathcal{A}$ can be embedded into some future clump. Note that if a clump $\mathcal{C}_i \hookrightarrow \mathcal{C}_j$, and if we have built a computable copy of $\mathcal{C}_i$, then we can extend this to a computable copy of $\mathcal{C}_j$ by adding new numbers and defining h and $<$ on them (without changing any decisions already made for numbers in the copy of $\mathcal{C}_i$).

We will construct $\mathcal{B} = (\omega, \leq_{\mathcal{B}}, h^{\mathcal{B}})$ which is isomorphic to $\mathcal{A}$ but not computably isomorphic to it. At stage s, we will have a finite structure $\mathcal{B}_s$, and a computable partial isomorphism f_s mapping an initial segment of $\mathcal{A}$ to $\mathcal{B}_s$. We let $\mathcal{C}_{i,s}^{\mathcal{B}}$ denote the image of $\mathcal{C}_i$ under f_s. The construction will ensure that f_s and f_s^{-1} are only redefined finitely often on any fixed value, so that $f = \lim_s f_s$ exists and is an isomorphism. We will also meet the following requirements for each e:

$$R_e : \varphi_e \text{ is not an isomorphism from } \mathcal{B} \text{ to } \mathcal{A}.$$

We will meet this requirement by inserting clumps into $\mathcal{B}$ whenever φ_e threatens to define the unwanted isomorphism.

At stage $s = 0$, just let $\mathcal{B}_0 = \mathcal{C}_0$, and $f_0 : \mathcal{C}_0 \to \mathcal{C}^{\mathcal{B}}_{0,0}$ be the identity. At stage $s+1$, look for the least e such that φ_e is an isomorphism taking $\mathcal{B}_s$ to $\mathcal{A}$ as far as a clump $\mathcal{C}^{\mathcal{B}}_{n,s}$ that is free for R_e. If no such e exists, let f_{s+1} and $\mathcal{B}_{s+1}$ extend the partial isomorphism f_s to the next clump of $\mathcal{A}$. Otherwise, begin to define $\mathcal{B}_{s+1}$ as follows. The initial segment of $\mathcal{B}_{s+1}$ agrees with $\mathcal{B}_s$ on all clumps $\mathcal{C}^{\mathcal{B}}_{l,s}$ with $l < n$. Now insert into $\mathcal{B}_{s+1}$ another copy of $\mathcal{C}_n$ immediately preceding $\mathcal{C}^{\mathcal{B}}_{n,s}$. Then we have a partial isomorphism f_{s+1} mapping the first n clumps of $\mathcal{A}$ to $\mathcal{B}_{s+1}$ which disagrees with φ_e. We now ensure that $\mathcal{B}_{s+1}$ is isomorphic to an initial segment of $\mathcal{A}$. Let $m \geq n$ be such that $\mathcal{B}_s = \mathcal{C}^{\mathcal{B}}_{0,s} < ... < \mathcal{C}^{\mathcal{B}}_{n,s} < ... < \mathcal{C}^{\mathcal{B}}_{m,s}$. Let $t_{n-1} = n$. For $n \leq l \leq m$, do as follows. Look for the first clump $\mathcal{C}_{t_l}$ after $\mathcal{C}_{t_{l-1}}$ such that $\mathcal{C}_l$ embeds into $\mathcal{C}_{t_l}$, and insert the segment $\mathcal{C}_{t_{l-1}+1}, \ldots, \mathcal{C}_{t_l-1}$ into $\mathcal{B}_{s+1}$ between $\mathcal{C}^{\mathcal{B}}_{t_{l-1},s+1}$ and $\mathcal{C}^{\mathcal{B}}_{l,s}$. Expand $\mathcal{C}^{\mathcal{B}}_{l,s}$ to a copy of $\mathcal{C}_{t_l}$. This will extend f_{s+1} so that the domain includes $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{t_m}$. Declare all these clumps $\mathcal{C}^{\mathcal{B}}_{0,s+1}$,...,$\mathcal{C}^{\mathcal{B}}_{t_m,s+1}$ to not be free for any R_i with $i > e$. The structure $\mathcal{B}$ is the limit of the structures $\mathcal{B}_s$. $\mathcal{B}$ is computable since the ordering $\leq^{\mathcal{B}}$ and the function $h^{\mathcal{B}}$ are defined on new elements as soon as they come into $\mathcal{B}$ and never change. The isomorphism $f : \mathcal{A} \to \mathcal{B}$ is the limit of the sequence $\{f_s : s < \omega\}$. Each requirement R_e only causes a finite shift in $\mathcal{B}_s$, and only causes finitely many clumps to not be free for lower priority requirements. Thus each requirement is eventually satisfied, either because φ_e never defines an isomorphism as far as the first clump that is free for R_e, or because we inserted some clump to defeat a possible isomorphism at a stage where R_e had obtained the highest priority. Our marking of clumps as "not free" for lower priority requirements once some requirement has caused them to shift ensures that each clump of $\mathcal{B}$ can be shifted only finitely often once it has been introduced, and hence the map f is indeed an isomorphism. □

Now we look at the case where there is no infinite trace, but there is also no bound on the length of traces.

Theorem 3.4. *There exists a structure $(\omega, \leq, h)$ where the traces are all finite but the structure is computably categorical.*

In order to prove the above theorem, we will make use of *special clumps* that cannot be embedded into one another. Since these will also be useful in later theorems, we define them now.

Definition 3.5. For each $i \in \omega$ let $\mathcal{D}_i = (D_i, \leq, h)$ be a structure with domain $D_i = \{x_0, x_1, ..., x_{i+1}\}$, $x_j < x_{j+1}$ for $0 \leq j \leq i$, $h(x_0) = x_2$, $h(x_j) = h(x_{j+1})$ for $1 \leq j \leq i$, and $h(x_{i+1}) = x_{i+1}$.

Note that for $i \neq j$, $\mathcal{D}_i \not\hookrightarrow \mathcal{D}_j$.

Proof of Theorem 3.4. Define $h : \omega \to \omega$ such that $(\omega, \leq, h)$ is isomorphic to the order $\mathcal{D}_0 < \mathcal{D}_1 < ....$ Now suppose $\mathcal{A} \cong (\omega, \leq, h)$. To define the computable isomorphism, just enumerate the approximation to $\mathcal{A}$. Since the $\mathcal{D}_i$ cannot embed into one another, we know that once something isomorphic to $\mathcal{D}_i$ shows up in $\mathcal{A}$, it must be mapped to $\mathcal{D}_i$ in $(\omega, \leq, h)$. A clump in $\mathcal{A}$ can be identified as a sequence $a_0 < a_1 < \cdots < a_i$ such that $h^{\mathcal{A}}(a_0) = a_2$, $h^{\mathcal{A}}(a_j) = a_{j+1}$ for $j < i$ and $h^{\mathcal{A}}(a_i) = a_i$. Given $a \in \mathcal{A}$, simply wait for a clump of size i containing a and then map a into $\mathcal{D}_i$. □

Theorem 3.6. *Let $X = \{i \mid (\forall j > i)\mathcal{C}_i \not\hookrightarrow \mathcal{C}_j\}$. If X is not h-immune then the order is computably categorical.*

Proof. Suppose X is not hyperimmune. Then there exists a computable function g that majorizes the principal function of X. That is, among $\mathcal{C}_0, ..., \mathcal{C}_{g(n)}$ there are at least n-many clumps that cannot be embedded into any later clumps. Thus given an approximation of $\mathcal{A} \cong (\omega, \leq, h)$, we may define a computable isomorphism as follows. To define the isomorphism on the initial segment $\mathcal{C}_0 < ... < \mathcal{C}_n$ of $(\omega, \leq, h)$, run the approximation of $\mathcal{A}$ until we see an initial segment isomorphic to $\mathcal{C}_0 < ... < \mathcal{C}_{g(n)}$. Then since this initial segment contains n-many clumps that cannot embed into future clumps, it must be correct on at least the first n clumps. So we can define the isomorphism accordingly. □

However, the converse is not true.

Theorem 3.7. *There exists a structure $(\omega, \leq, h)$ with X hyperimmune and $(\omega, \leq, h)$ computably categorical.*

Proof. We will build $(\omega, \leq, h)$ using the special clumps $\mathcal{D}_i$ from Definition 3.5. The idea will be that as we see possible computable copies that appear to be isomorphic to the $(\omega, \leq, h)$ that we are building, we use the fact that the special clumps cannot embed into one another to force a unique (computable) isomorphism. To meet the hyperimmunity requirement, we ensure that non-repeated special clumps occur sparsely.

We build the structure by stage by stage revealing a longer initial segment of it, always adding entire clumps, and only to the end. That is, if at stage s we have $\mathcal{A}_s = \mathcal{C}_0 < ... < \mathcal{C}_n$ then at stage $s + 1$ we will extend this by adding clumps after $\mathcal{C}_n$ to obtain $\mathcal{A}_{s+1} = \mathcal{C}_0 < ... < \mathcal{C}_n < \mathcal{C}_{n+1} < ... < \mathcal{C}_{n+m}$. Let requirement R_e work for building a computable isomorphism between $\mathcal{A}$ and the structure being built by Ψ_e, if it is isomorphic to $\mathcal{A}$. Each R_e requirement will hold two special clumps. The two special clumps may not be repeated. When Ψ_e looks the same as $\mathcal{A}$ up to the second special clump, we make this the first special clump, choose a new second special clump, and allow the original first special clump to be repeated. We extend the definition of the isomorphism up to the new first special clump. This will have to be correct, since in order to change the initial segment that Ψ_e has revealed and still be isomorphic to $\mathcal{A}$, there must be a place to absorb the first special clump. We will not provide such a place if Ψ_e has changed.

To make X h-immune: When $\varphi_e(e)\downarrow$, ensure that there are at most e special clumps less than $\varphi_e(e)$. We do this by extending the initial segment of $\mathcal{A}$ by repeating all but e of the clumps that are less than $\varphi_e(e)$. This will injure requirements R_i for $i < \frac{e}{2}$, and we will have to completely restart our definition of the isomorphism for those R_i. However, each R_i will be injured at most $2i$-times, after which, if $\Psi_i \cong \mathcal{A}$, then it's computable isomorphism will be total.

Stage 0: Let $\mathcal{A}_0 = \mathcal{D}_0 < \mathcal{D}_1$. Set $c_0(0,0) = 0$ and $c_0(0,1) = 1$.

Odd stage s: Suppose $\mathcal{A}_s = \mathcal{C}_0 < .... < \mathcal{C}_n$. If $\varphi_{e,s+1}(e)\downarrow$, then for each $i < \varphi_e(e)$ such that $i \neq c_s(j,k)$ for any $j < \frac{e}{2}$, add a copy of $\mathcal{C}_i$ to $\mathcal{A}_{s+1}$. For each such i, declare f_i injured. Also, plunk down new, unused special clumps, and define $c_{s+1}(j,k)$ for $\frac{e}{2} \leq j \leq s+1$. Set $c_{s+1}(j,k) = c_s(j,k)$ for $j < \frac{e}{2}$.

Even stage s: Suppose an initial segment of Ψ_e appears to be isomorphic to our initial segment of $\mathcal{A}_s$ up to $c_s(e,1)$. Then extend the definition of f_e to map the initial segment of $\mathcal{A}_s$ up to $c_s(e,1)$ to the corresponding initial segment of Ψ_e. Let $c_{s+1}(e,0) = c_s(e,1)$. Suppose $\mathcal{A}_s = \mathcal{C}_0 < ... < \mathcal{C}_n$. Let j be least such that $\mathcal{D}_j \neq \mathcal{C}_i$ for $i \leq n$. Let $\mathcal{A}_{s+1} = \mathcal{A}_s < \mathcal{D}_j$, and let $c_{s+1}(e,1) = n+1$.

Lemma 3.8. *X is hyperimmune.*

Proof. We must show that for every computable function g, there exist infinitely many $n \in \omega$ such that $g(n) \geq p_X(n)$. If g is computable, then there exist infinitely many n such that $\varphi_n = g$. By construction, $\varphi_n(n) \geq p_X(n)$. □

Lemma 3.9. *If $\Psi_e \cong \mathcal{A}$ then f_e is a total computable function and $f_e : \mathcal{A} \cong \Psi_e$.*

Proof. Suppose that for all $n \leq 2e$, $\varphi_n(n)\downarrow \Rightarrow \varphi_{n,s}(n)\downarrow$. Then $c_t(e,i)$ will only be redefined at odd stages after stage s. Let $t_0 > s$. Since $\Psi_e \cong \mathcal{A}$, there must have been a stage $t_1 \geq t_0$ where Ψ_e appeared isomorphic to $\mathcal{A}$ up to $\mathcal{C}_{c_{t_0}(e,1)}$. At that moment we would have defined f_e up to $\mathcal{C}_{c_{t_0}(e,1)}$, and set $c_{t_1}(e,0) = c_{t_0}(e,1)$. Since $\Psi_e \cong \mathcal{A}$, there must have been a stage $t_2 \geq t_1$ where Ψ_e appeared isomorphic to $\mathcal{A}$ up to $\mathcal{C}_{c_{t_1}(e,1)}$. This isomorphism must have extended the isomorphism that was observed at stage t_0, because there would be nowhere to absorb $\mathcal{C}_{c_{t_1}(e,0)}$. □

This completes the proof of Theorem 3.7.

□

References

[1] W. Calvert, D. Cenzer, V. Harizanov, and A Morozov. Effective categoricity of equivalence structures. *Ann. Pure Appl. Logic*, 141:61–78, 2005.

[2] Peter Cholak, Sergey Goncharov, Bakhadyr Khoussainov, and Richard A. Shore. Computably categorical structures and expansions by constants. *J. Symbolic Logic*, 64(1):13–37, 1999.

[3] V. D. Dzgoev and S. S. Goncharov. Autostability of models. *Algebra i Logika*, 19:45–58, 1980.
[4] S. S. Goncharov. The problem of the number of nonautoequivalent constructivizations. *Algebra i Logika*, 19(6):621–639, 745, 1980.
[5] B. M. Khusainov. The algorithmic dimension of unars. *Algebra i Logika*, 27(4):479–494, 499, 1988.
[6] Jeffrey B. Remmel. Recursively categorical linear orderings. *Proc. Amer. Math. Soc.*, 83:387–391, 1981.
[7] Jeffrey B. Remmel. Recursively rigid Boolean algebras. *Ann. Pure Appl. Logic*, 36(1):39–52, 1987.
[8] Yu. G. Ventsov. Algorithmic properties of branching models. *Algebra i Logika*, 25(4):369–383, 494, 1986.

Department of Mathematics, P.O. BOX 118105 University of Florida, 358 Little Hall, Gainesville, FL 32611
URL: www.math.ufl.edu/∼cenzer
E-mail address: cenzer@math.ufl.edu

Department of Pure Mathematics, University of Waterloo, Waterloo, ON, Canada N2L 3G1
URL: www.math.uwaterloo.ca/∼csima
E-mail address: csima@math.uwaterloo.ca

Department of Computer Science, University of Auckland, Auckland Private Bag 92019 Auckland New Zealand
URL: www.cs.auckland.ac.nz/∼bmk
E-mail address: bmk@cs.auckland.ac.nz