ON A QUESTION OF KALIMULLIN

ROD DOWNEY, GREGORY IGUSA, AND ALEXANDER MELNIKOV

Abstract. We prove that for every computable limit ordinal α there exists a computable structure $\mathcal{A}$ which is Δ^0_α-categorical and α is smallest such, but nonetheless for every isomorphic computable copy $\mathcal{B}$ of $\mathcal{A}$ there exists a $\beta < \alpha$ such that $\mathcal{A} \cong_{\Delta^0_\beta} \mathcal{B}$. This answers a question raised by Kalimullin in personal communication with the third author.

1. Introduction

Much of classical mathematics is concerned with classification of mathematical structures by their isomorphism types. Two mathematical structures are usually identified if they are isomorphic. However, such a classification blurs fine-grained distinctions related to the algorithmic nature of the structures. For example, it is easy to construct two algorithmically presented structures, both isomorphic to a simple structure like $(\mathbb{N}, \leq)$ but with wildly differing computability-theoretic properties, such as decidability (or non-decidability) of the adjacency relation. On the other hand, sometimes computable isomorphism type coincides with classical isomorphism type, which is the case for a dense countable linear ordering or a finitely presented group.

Computable structure theory [AK00, EG00] has grown to understand the computability-theoretic properties of *computably presented* structures. Recall that a countably infinite algebraic structure is *computable* if its domain is the natural numbers $\mathbb{N}$ and its operations are Turing computable functions. If an algebraic structure $\mathcal{A}$ is isomorphic to a computable structure $\mathcal{B}$, then we say that $\mathcal{B}$ is a computable copy, a computable presentation, or a constructivization of $\mathcal{A}$. As we noted above, an algebraic structure may have computable copies with wildly differing computability-theoretic properties. Based on this observation, Maltsev [Mal61] suggested that computable structures should be studied under computable isomorphism. In particular, we say that a countably infinite structure is *computably categorical* if it has a unique computable copy up to computable isomorphism. Although computably categorial structures are "unclassifiable" in general (see [DKL$^+$15]), computable categoricity tends to be very well-behaved within many

standard algebraic classes. For instance, a Boolean algebra is computably categorical iff it has only finitely many atoms [Gon97, LaR77], and a torsion-free abelian group is computably categorical iff its rank is finite [Nur74]. For most of these "nice" algebraic classes, computable categoricity is equivalent to the stronger notion of relative computable categoricity; we omit the definition of relative computable categoricity, see [AK00]. In contrast to computable categoricity in general, relative computable categoricity admits a syntactical description, a so-called computably enumerable Scott family [AK00]. There have been many successful applications of various syntactical techniques to the study of relative computable categoricity and related notions, see e.g. the recent work of Montalban [Mon13, Mon15] that relate computable structure theory with descriptive set theory and model theory. On the other hand, the study of the more "wild" general computable categoricity enjoys applications of advanced recursion-theoretic techniques; it also often leads to novel methods and new results which are not necessarily related to categoricity questions (see e.g. [DM13, Gon81]). One of the most remarkable theorems of this kind says that there is a structure with exactly two computable copies up to computable isomorphism; see Goncharov [Gon80], and see Hirschfeldt [Hir01] for further applications of the technique of Goncharov.

As we see, there are two main strands within modern computable structure theory. The first seeks to relate definability with effectivity [AK00], and the other tends to be concerned with properties which revolve around the specifics of the computation level and the structures concerned, see [EG00]. In this paper, we will be working on the interaction between the two strands. More specifically, we answer the following question of Kalimullin. A few years ago, Kalimullin asked whether a computable structure could be arithmetically categorical in the following unbounded way. Can there be a computable structure $\mathcal{A}$, such that for any computable structure $\mathcal{B}$ isomorphic to $\mathcal{A}$, there is an n such that $\mathcal{A}$ is Δ^0_n-isomorphic to $\mathcal{B}$, but for each $m < \omega$ there is a computable $\mathcal{C}_m$ with $\mathcal{C}_n$ isomorphic to $\mathcal{A}$ but *not* by a Δ^0_m-isomorphism? In other words, can there be a computable structure such that every isomorphic computable structure is arithmetically isomorphic, but such that this fact is not witnessed by any fixed level of the arithmetic hierarchy? In this paper we answer this question affirmatively. In fact, we prove more.

Theorem 1.1. *For every computable limit ordinal α there exists a computable structure $\mathcal{A}_\alpha$ such that:*

- *For every computable structure $\mathcal{M}$, there exists a $\beta < \alpha$ such that $\mathcal{M} \cong_{\Delta^0_\beta} \mathcal{A}_\alpha$.*
- *For every $\beta < \alpha$, there exists a computable structure $\mathcal{B} \cong \mathcal{A}_\alpha$ such that $\mathcal{B} \ncong_{\Delta^0_\beta} \mathcal{A}_\alpha$.*

We note that the structure $\mathcal{A}_\alpha$ witnessing Theorem 1.1 will be built up from structures that are themselves relatively Δ^0_β-categorical. Although the isomorphism types of these substructures will depend on the construction, their nice uniform properties will allow us to exploit techniques borrowed from the "syntactical" strand, more specifically a result of Ash [Ash86]. The construction will be an iterated priority argument, along the lines of a "worker" argument. We believe that the easiest presentation is to give a direct proof rather than to try to use any of the existing metatheorems ([AK00, Mon14, LL97]), aside from using the above mentioned result of Ash[1]. The remainder of this paper will be focused on proving Theorem 1.1.

2. Proof

2.1. **Setup and notation.** Let α be a fixed computable limit ordinal. For notational convenience, when we wish to discuss Δ^0_β constructions as oracle constructions, we will use $0_{(\beta)}$ as the name of the oracle which is equal to $0^{(\beta-1)}$ when $1 \leq \beta < \omega$, and is equal to the β'th jump (using specifically chosen Turing degree representatives that can uniformly resolve Δ^0_β questions) when $\omega \leq \beta < \alpha$.

Let $\langle \alpha_n : n \in \omega \rangle$ be a computable increasing sequence of ordinals whose limit is α, with $\alpha_0 > 0$. For each n, let $\beta_n = 2 \cdot \alpha_n + 1$, and note that $\langle \beta_n : n \in \omega \rangle$ is also a computable increasing sequence of ordinals whose limit is α, but with $\beta_0 > 2$.

The signature of our structure will have an edge relation, an ordering relation, and a collection of unary predicates $P_{n,i}$, denoting disjoint portions of the structure which we will use to meet diagonalization requirements. We will call the part of the structure restricted to one such $P_{n,i}$ a P-box. The ordering relation and edge relation will never hold between elements of different P-boxes.

2.1.1. *The requirements.* Let $\langle \mathcal{M}_n : n \in 1, 2, \ldots \rangle$ be a listing of all the computable structures in our signature, which will be specified later.

[1]Thanks to Noam Greenberg, we are aware that Asher Kach and Antonio Montalban have (independently) announced the case $\alpha = \omega$. As far as we know, their proof has not yet been formally written or published.

We will have isomorphism requirements and diagonalization requirements:

$$\mathcal{I}_n : \mathcal{M}_n \cong \mathcal{A} \to \mathcal{M}_n \cong_{\Delta^0_{\beta_{n-1}+1}} \mathcal{A}$$

$$\mathcal{D}_n : \exists \mathcal{B}_n (\mathcal{B}_n \text{ is computable } \& \ \mathcal{B}_n \cong \mathcal{A} \ \& \ \mathcal{B}_n \ncong_{\Delta^0_{\beta_n}} \mathcal{A})$$

Note that if we meet all of these requirements, then we will have proved the theorem. (For trivial counting reasons such as β_{n-1} above and without loss of generality, we may assume that n ranges over the positive natural numbers.)

We will discuss two different diagonalization strategies. The "attempted diagonalization strategy" will be the naive attempt of meeting a $\mathcal{D}_n$ requirement, in isolation. The "actual diagonalization strategy" will be different from the attempted one due to interactions between requirements. The isomorphism-building strategy $\mathcal{I}_n$ will be formally described only after the tree of strategies is introduced, but we will give some intuition already in the next subsection.

2.2. **One diagonalization strategy in isolation.** In this subsection, we describe the attempted diagonalization strategy that will have to be modified before placing it onto the tree.

2.2.1. *The result of Ash.* Our primary tool for the attempted diagonalization strategy will be the following result of Ash (Theorem 18.15 of [AK00]).

Theorem 2.1 (Ash)**.** *Let γ be a computable ordinal, and suppose L is a $\Delta^0_{2\gamma+1}$ linear ordering. Then we can uniformly produce a computable presentation $F(L)$ of $\omega^\gamma \cdot L$.*

It will be crucial that the proof of Theorem 2.1 is uniform as long as L does not have a least element. Moreover, there is a $\Delta^0_{2\gamma+1}$ function f taking $a \in L$ to the first element of the corresponding copy of ω^γ in $F(L)$. The $\Delta^0_{2\gamma+1}$ index of the function is also uniform in (the notation for) γ and the index for L.

Remark 2.2. Using the uniformity of Theorem 2.1 we will shortly define our attempted diagonalization strategy on $P_{n,i}$. In the proof of the main result, all such $P_{n,i}$ will be put together using sequences of guesses. All such sequences (in essence) form is a degenerate priority tree. At every node of the tree, we will use the uniform version of Theorem 2.1 to produce copies of linear orders (they will be slightly modified, but with all possible uniformity). One could use the recursion theorem to make the tree work.

We remark here that the recursion theorem is not really necessary here. Each local version of the construction from Theorem 2.1 will be used by its node without essential modifications. It can be paused, then perhaps restarted, or perhaps

permanently terminated, depending only on our current guess. The $\Delta^0_{2\gamma+1}$-index of the input order L will also depend only on our current guess and some elementary actions, all being uniform. We simply dynamically outsource the task of building $F(L)$ to the construction from Theorem 2.1 whose index is known (by the s-m-n Theorem).

For our attempted diagonalization strategy on $P_{n,i}$, we begin by constructing two linear orders in $\mathcal{A}$ and $\mathcal{B}_n$, that we will ensure are isomorphic, but not a $\Delta^0_{\beta_n}$ isomorphism. This will be done by constructing linear orderings L_0 and L_1, computably in $0_{(\beta_n)}$, both of order type ω^*, and then constructing $F(L_0)$ as our linear ordering in $P_{n,i}$ of $\mathcal{A}$ and $F(L_1)$ as our linear ordering in $P_{n,i}$ of $\mathcal{B}_n$. Here, we are using Theorem 2.1 with $\gamma = \alpha_n$, and hence $2\gamma + 1 = \beta_n$.

2.2.2. *The construction of $F(L_0)$ and $F(L_1)$.* We build two $\Delta^0_{\beta_n}$-copies of ω^*. We diagonalize against the $e'th$ potential $\Delta^0_{\beta_n}$ isomorphism $f_e : L_0 \to L_1$, as follows. Construct ω^* in both L_0 and L_1, initially letting $x_{0,e}$ and $x_{1,e}$ be adjacent elements in L_0. Wait for f_e to converge on $x_{0,e}$ and $x_{1,e}$. If the images are adjacent, insert one extra point between them, and preserve the interval. Note that this construction is computable in $0_{(\beta_n)}$. Both L_0 and L_1 have no least element, so that we can apply Theorem 2.1 with all possible uniformity. In particular, we obtain $F(L_0)$ and $F(L_1)$ which are isomorphic, but not by a $\Delta^0_{\beta_n}$-map.

For each element of our linear orders $F(L_0)$ and $F(L_1)$, we will have one extra element that is attached to it by the edge relation. The additional elements will not be related to any elements via the ordering relation.

Remark 2.3. At this stage we owe the reader an informal explanation of why we need this extra edge relation. The idea is as follows. Suppose at stage s our linear order that we've build in $P_{n,i}$ of $\mathcal{A}$ looks as follows:

$$\widehat{a}_0 < \widehat{a}_1 < \widehat{a}_3 < \widehat{a}_4,$$

and we also have build a partial map from the opponent's structure $f_s : \mathcal{M}_s \to \mathcal{A}_s$. The hat indicates that a point carries a label, i.e., it is adjacent to an extra point via the edge relation. (The ith extra point is specific and unique for each such a_i and is not connected to anything else.) According to our dynamic definition of the linear order in $P_{n,i}$, we will attempt to extend to $P_{n,i}$ by adding one more point, say b. We add the point without a label:

$$\widehat{a}_0 < b < \widehat{a}_1 < \widehat{a}_3 < \widehat{a}_4,$$

and then wait for $\mathcal{M}$ to respond. Note the location of b is uniquely determined by the configuration at the previous stage s. If $\mathcal{M}$ gives us some other configuration, we permanently "kill" $\mathcal{M}$ by, say, freezing our enumeration in $P_{n,i}$. In particular, if $\mathcal{M}$ is too quick in its enumeration then it will be "killed". As soon as we see a

suitable candidate for $f^{-1}(b)$ (if ever), we extend f_s to b. Only then we put a label onto b:

$$\widehat{a}_0 < \widehat{b} < \widehat{a}_1 < \widehat{a}_3 < \widehat{a}_4,$$

and wait for $f^{-1}(b)$ to be labeled in $\mathcal{M}$ as well. Finally, we then extend f to that label. Repeat. This way we force $\mathcal{M}$ to follow us very closely. At the end $\mathcal{M}$ is either dead (i.e., not isomorphic to $\mathcal{A}$) or is forced to copy us via the *computable* isomorphism f. (W.l.o.g., we assume from the beginning that the domain of $\mathcal{M}$ is a c.e. subset of ω.) In $\mathcal{B}_n$, we don't have to do anything too carefully. We just build a copy of $F(L_1)$ by stages, unless interrupted.

In $\mathcal{A}$ we will later need to implement the idea from Remark 2.3. However, in isolation the construction of $P_{n,i}$ in both $\mathcal{A}$ and $\mathcal{B}_n$ look the same (with L_0 replaced by L_1), as follows.

2.2.3. *The attempted $\mathcal{D}_n$ strategy.* The way that we construct this structure in $\mathcal{A}$ in stages is as follows. During the first stage, we put the first two elements into $F(L_0)$, and we attach an element to each of them.

After this, each time we wish to add a new element to $F(L_0)$, we will do so over the course of two substages. At the beginning of the first substage, there will be a finite number of elements in $F(L_0)$, each with an element attached by an edge.

Substage 1: Add a new element x to $F(L_0)$.

Substage 2: Create a new element and attach by an edge to x.

At the end of the construction, we will have constructed $F(L_0)$ with an extra element attached to each element of $F(L_0)$. (This ends the strategy.)

The lemma below if fairly straightforward. As we have already noted above, the actual strategy may be interrupted by higher priority requirements, and thus may never finish building its $P_{n,i}$ block in $\mathcal{A}$ and $\mathcal{B}_n$. The lemma below describes what happens in absence of such interaction, i.e. when it acts as intended.

Lemma 2.4. *The attempted diagonalization strategy on $P_{n,i}$ in $\mathcal{A}$ satisfies the following properties.*

1. *The attempted diagonalization strategy is uniform (in n, i).*
2. *If the attempted diagonalization strategy is completed, then $\mathcal{A}$ is not isomorphic to $\mathcal{B}_n$ via a $\Delta^0_{\beta_n}$ map.*
3. *If the attempted diagonalization strategy is completed, then $P_{n,i}$ of $\mathcal{A}$ is relatively $\Delta^0_{\beta_n+1}$-categorical.*

Proof. (1) The uniformity follows from the fact that the sequence $\langle \alpha_n \rangle$ is computable, as well as the fact that the proof of Theorem 2.1 is uniform. (Also, see Remark 2.2 above.)

(2) Any isomorphism between $\mathcal{A}$ and $\mathcal{B}_n$ must also be an isomorphism when restricted to $P_{n,i}$, and must therefore also be an isomorphism when restricted to the linear order part of $P_{n,i}$. In our construction, we ensured that $F(L_0)$ and $F(L_1)$ were not isomorphic by the ith Δ^0_β map, for each i.

(3) This folklore fact follows at once from a straightforward definability analysis (see the book [AK00]) □

The properties of the attempted diagonalization strategy (e.g., Lemma 2.4(2)) ensure that if for every n and i there exists a j such that $P_{n,i}$ completes its attempted construction, then all of the $\mathcal{D}_n$ requirements will be met.

2.3. **Construction.** As we noted above, the attempted diagonalization strategy needs to be modified in the actual construction. We have already discussed the idea in Remark 2.3. In this subsection we give the formal details.

2.3.1. *Informal discussion.* The construction will be phrased as a tree construction. We note that the tree itself will be a standard computable tree in which the true path can be approximated in the usual Π^0_2-fashion. Each node will have two outcomes, ∞ and fin. The outcomes at level n will reflect our current guess on whether the nth partial computable structure has followed us for one more step or not, in the sense of Remark 2.3. (This can be formally implemented using, e.g., expansionary stages.) Since there is nothing special about this particular tree, we will not waste our time formally defining the true path, expansionary stages etc. This is left to the reader.

What is unusual in this proof is that each node at level n will be associated with a copy of the diagonalization strategy, and thus will be working relative to $0_{(\beta_n)}$ within its P-box. Using the construction of Ash (Theorem 2.1), we uniformly extend the $0_{(\beta_n)}$-linear order produced by the strategy to a computable linear order.

2.3.2. *The tree of strategies.* The tree of strategies is $\{\infty, fin\}^{<\infty}$. Each level of the tree will be monitoring the nth partial computable structure $\mathcal{M}_n$. Each node at this level will attempt to build a *computable* partial isomorphism on the P-boxes (in $\mathcal{A}$ and $\mathcal{M}_n$) which are controlled by requirements of priorities no stronger than the priority

of σ. This will be done as follows. Every time σ or any weaker priority requirement below $\sigma\hat{}\infty$ puts a new point into their part of $\mathcal{A}$, we wait for $\mathcal{M}_n$ to respond by giving us the exact same configuration, restricted to this part. (Clearly, the node σ cannot hope to force the higher priority $\mathcal{M}_k$, $k < n$.) We will clarify later that the actions of σ

2.3.3. *Assignment of P-boxes to strategies.* Recall that each such strategy is working within its P-box. If σ is at level n, then at stage s at which we access it the first time we reserve $P_{n,i}$ (with i least never used for this purpose) for the infinitary action of σ. We denote this P-box of σ by P^∞_σ. We will be pressing $\mathcal{M} = \mathcal{M}_n = \mathcal{M}_{|\sigma|}$ to follow $\mathcal{A}$ computably within P^∞_σ by pausing our actions within P^∞_σ of $\mathcal{A}$ (thus, of $\mathcal{B}$ as well) until the P^∞_σ-box of $\mathcal{M}$ is extended to be isomorphic to P^∞_σ of $\mathcal{A}$ built so far. We will give formal details of the modified diagonalization strategy in the next paragraph, but we have already seen the main idea in Remark 2.3. Since $\mathcal{M}$ may never respond, we have to restart the diagonalization strategy within some $P_{n,j}$ (where j is least never used so far by nodes at level n). We denote this box by P^{fin}_σ. Within this box, we will try to implement the attempted diagonalization strategy ignoring $\mathcal{M}$, until (if ever) $\mathcal{M}$ responds on P^∞_σ or the current true path moves to the left of σ. In the former case, we initialize only P^{fin}_σ, in the latter case we initialize both P^∞_σ and P^{fin}_σ, as will be described below shortly. Next time σ is visited again it will pick a new fresh pair of P-boxes. (We can arrange the construction so that no P-box can be ever left unattended at the end.)

2.3.4. *The actual diagonalization strategy.* Each P-box will be eventually assigned to a node on the tree, suppose P is assigned to σ. Each such box will be following the attempted diagonalization strategy, but with one important modification. Between Substage 1 and Substage 2 (see the previous section) it will be waiting for every $\mathcal{M}_{|\tau|}$ with $\tau\hat{}\infty \subseteq \sigma$ to reveal x. (If P is of the form P^∞_σ it will also wait for $\mathcal{M}_{|\sigma|}$.) After Substage 2 is finished, it will also wait for each such $\mathcal{M}_{|\tau|}$ to reveal the extra element now attached to x by an edge. (Note that this pause also applies to the respective P-box of $\mathcal{B}_{|\sigma|}$.)

2.3.5. *Initialization.* Suppose we need to initialize a P-box (denote it simply by P) for one of the reasons described above. This is done as follows. Currently P contains a finite linear order, at most one element

of which is labeled (using the edge relation). We leave this box forever unattended. The finite structure within it will never be changed again.

As we noted above, the definition of the current true path is standard. At stage s of the construction, we simply let the strategies along the current true path act according to their instructions.

2.4. **Verification.** The intuition is as follows. We will argue that $\mathcal{M} \cong \mathcal{A}$ implies that the true outcome of the node σ at level n of the true path is ∞. The strategy of σ cannot control the finitely many nodes above it, but for σ this is just a finite noise. For each strategy τ at deeper levels of the tree, there are only two possibilities. It will either be eventually left finite, in which case their box will be finite and thus uniformly Δ^0_2-categorical. Otherwise, each box ever controlled by τ will have to respect $\mathcal{M}_n$ as well as the structure $M_{|\tau|}$. Note that $0''$ can see which boxes are initialized, since it can compute the true path. Thus, σ can ensure that almost all P-boxes of $\mathcal{M}$ are uniformly $\Delta^0_{\beta_{n-1}+1}$-isomorphic to the respective boxes of $\mathcal{A}$ (see Lemma 2.4(3)).

To the details. We now verify that the construction as described above will meet all of the $\mathcal{D}_n$ requirements as well as all of the $\mathcal{I}_n$ requirements. Although the actions of each individual diagonalization strategy could be computationally relatively complex, the guessing procedure that determines the current true path is merely Π^0_2. It is clear that $\mathcal{M}_{|\sigma|}$ can be either isomorphic to $\mathcal{A}$ or not isomorphic to $\mathcal{A}$. (The same can be said if we restrict our guessing to a computable collection of P-boxes.) The definitions of the true path and the true outcome are standard. We conclude that there are only two possible (true) outcomes of the guessing procedure. The tree does not depend on our diagonalisation actions whose outcomes are not even put onto the tree. It is thus straightforward that the true path is infinite.

We need to argue that, for every $\sigma\hat{}x$ along the true path (where $x \in \{\infty, fin\}$), the diagonalization requirement $\mathcal{D}_{|\sigma|}$ is met within P^x_σ (and, thus, for the whole $\mathcal{A}$ and $\mathcal{B}$). We also need to check that, if $M_{|\sigma|}$ is isomorphic to $\mathcal{A}$ then it is isomorphic to $\mathcal{A}$ via an isomorphism strictly computationally simpler than $0_{(\alpha)}$.

Lemma 2.5. *For every n, the diagonalization requirement $\mathcal{D}_n$ is met within the P^x_σ-box, where $\sigma\hat{}x$ is along the true path and $|\sigma| = n$.*

Proof. For simplicity, we first consider the highest priority diagonalization requirement. If there were no higher priority $\mathcal{I}$-requirements

to respect, this would be exactly Lemma 2.4. However, according to our setup, $\sigma = e$ (the empty string) must respect $\mathcal{M}_0$. But if the true outcome of $\mathcal{I}_0$ is ∞, then $\mathcal{M}$ always responds by copying us at the intermediate stage, see the description of the actual diagonalization strategy at the previous section. Thus, the diagonalization strategy within P_e^∞ will be acting at infinitely many stages. Apart from pausing at the intermediate stage, the actual diagonalization strategy (see 2.3.4) is no different from the attempted one (see 2.2.3). Thus, we can appeal to Lemma 2.4 and conclude that $\mathcal{D}_0$ is met within the P_e^∞-box. On the other hand, if $\mathcal{M}$ eventually either never responds or proves to be non-isomorphic, then we implement the diagonalization strategy within some other, fresh box P_e^{fin} which will never be initialized. The strategy will ignore $\mathcal{M}_0$ and will be exactly the same as the attempted one (see 2.2.3). We thus can safely appeal to Lemma 2.4.

The general case of $n > 0$ is not very much different from the basic case $n = 0$. it is sufficient to take σ with $|\sigma| = n$ along the true path and consider the box P_σ^x, where x is the true outcome of σ. The only difference is that the strategy within P_σ^x will have to respect only those M_τ with $\tau\hat{}\infty \subseteq \sigma^x$. □

Lemma 2.6. *For every n, if $\mathcal{M}_n \cong \mathcal{A}$ then $\mathcal{M}_n \cong_{\Delta^0_{\beta_{n-1}+1}} \mathcal{A}$. (That is, $\mathcal{I}_n$ is met.)*

Proof. Consider $\mathcal{M}_n$ and assume that $\mathcal{M}_n \cong \mathcal{A}$. This is where we use the modification to the attempted diagonalization strategy (see 2.3.4). There are several types of P-boxes that we need to consider. We argue that in each case we can (uniformly) produce an isomorphism of complexity at most $\Delta^0_{\beta_{n-1}+1}$ between the respective boxes in $\mathcal{A}$ and $\mathcal{M}_n$.

Suppose $m < n$. Then $P_{m,i}$ is either eventually permanently assigned to P_τ^x for some τ of length m, or is eventually initialised (and thus is permanently left with a finite structure inside it). With the help of $0''$ we can see which case applies to any such $P_{m,i}$, $m < n$. By Lemma 2.4, $P_{m,i}$ of $\mathcal{A}$ is relatively $\Delta^0_{\beta_{m-1}+1}$-categorical. By the choice of the sequence of $(\beta_n)_{n\in\omega}$, $0''$ is no greater than each such β_m, thus the isomorphism within each $P_{m,i}$ is (uniformly) computable 0_{β_n}.

If $m \geq n$, then we should appeal to the intermediate stage of the actual diagonalization strategy (2.3.4). Again, each P-box is either eventually initialised or is stably controlled by one of the diagonalization strategies. As above, with the help of $0''$ we can see it. If the box is eventually initialized then this means that the substructure within it is finite, and thus $0''$ can uniformly and fully reconstruct its open

diagram. If it is never initialized, then suppose it is permanently declared P^x_τ for some τ of length $m \geq n$. Consider σ of length n along the true path. Then σ monitors $\mathcal{M}_n$, and it must be the case that the true outcome of σ is ∞. In partucular, $\sigma\hat{}\infty \subseteq \tau$. This means that the actual diagonalization strategy of P^x_τ will not add another point to the box unless $\mathcal{M}_\sigma$ responds by giving the previous point, see 2.3.4. At every such stage, P^x_τ will either be a finite linear order with at most one element not labeled. We will define a *computable* isomorphism between P^x_τ in $\mathcal{M}_m$ and P^x_τ in $\mathcal{A}$ by stages, as follows.

Let $P[s]$ denote the substructure of $\mathcal{A}$ restricted to the P^x_τ-box, at stage s. Similarly, $P'[s]$ will denote the respective substructure of $\mathcal{M}_n$, also at stage s. Suppose we have already defined an isomorphism $f_s : P[s] \to P'[s]$. We may assume that $P[s]$ is either empty (in which case f_s is also empty) or it is a finite linear order all of whose elements are labeled using the edge relation. (See 2.2.2 for the description of labels.)

At the next stage s^* at which τ is visited again, we will expand $P[s]$ by one extra point b to get $P[s^*]$. For now, we will keep this extra point not labeled. Here is a "typical" configuration within $P[s^*]$:

$$\widehat{a}_0 < \ldots < \widehat{a}_i < b < \widehat{a}_{i+1} < \ldots < \widehat{a}_{k(s)},$$

where only the elements that carry a hat are labeled. The hatted elements and the auxiliary elements that use to label them, form the domain of f_s. Since f_s is an isomorphism [restricted to its domain], in $\mathcal{M}_n[s^*]$ we have

$$\widehat{f_s(a_0)} < \ldots < \widehat{f_s(a_i)} < \widehat{f_s(a_{i+1})} < \ldots < \widehat{f_s(a_{k(s)})},$$

where the auxiliary elements labelling each of the $\widehat{f_s(a_i)}$ have the auxiliary elements labelling the respective $\widehat{a}_i$ as their f_s-preimages. After stage s^*, τ will enter the waiting phase (see 2.3.4). During this substage, it will be waiting for $\mathcal{M}_n$ (and perhaps for finitely many other structures as well) to respond. More formally, it will wait until $\mathcal{M}_n$ reveals a point c such that

$$\widehat{f_s(a_0)} < \ldots < \widehat{f_s(a_i)} < c < \widehat{f_s(a_{i+1})} < \ldots < \widehat{f_s(a_{k(s)})},$$

and so that it is currently not labeled. If $\mathcal{M}_n$ never responds or gives some other configuration, we will permanently abandon the infinitary outcome of σ. This will be done by simply freezing this P-box of $\mathcal{A}$. This way we will make sure $\mathcal{A} \not\cong \mathcal{M}_n$ contradicting the assumption. It is *crucial* that the configuration of labels and the linear order uniquely define the location of b. Thus, such a c must eventually be found in

$\mathcal{M}_n$, and it must necessarily be between $\widehat{f_s(a_i)}$ and $\widehat{f_s(a_{i+1})}$. Once c is found, at stage t, we extend f_s to f_t by setting $f_t(b) = c$.

After this is done, we add a label to b by introducing an auxiliary y and declaring $E(b, y)$ on it. This element y will not be related to any other element in the construction by $<$, and it will not be connected to anything else via the edge relation E. Thus, similarly to how we argued that c can be found for b, we could argue that a z in $\mathcal{M}_n$ can be found for y. It is also necessary that z labels c. We extend f accordingly.

Note that f is computable. Furthermore, f_s is an isomorphism of $P[s]$ onto $P'[s]$, for every s. It follows that f is a computable isomorphism from P^x_τ of $\mathcal{A}$ to P^x_τ in $\mathcal{M}_n$. The index of f can be found uniformly in $0''$.

We return to the proof of the lemma. We have argued that in every possible case we can, $\Delta^0_{\beta_{n-1}+1}$ -uniformly in i, n, find a $\Delta^0_{\beta_{n-1}+1}$ isomorphism between $P_{n,i}$-boxes in $\mathcal{A}$ and $\mathcal{M}_n$. □

We conclude that all requirements are met, and thus Theorem 1.1 is proved.

2.5. **Finding examples in nice classes.** From the algebraic standpoint, our structure $\mathcal{A}$ is an abomination. It was designed to prove the theorem. Can we find examples in nice classes?

Problem. Find structures with the property from Theorem 1.1 in natural non-universal classes, such as linear orders or abelian groups.

We strongly suspect that manufacturing examples of such linear orders should not be too hard. In contrast, we have no idea how to approach this question in the class of abelian groups.

References

[AK00] C. Ash and J. Knight. *Computable structures and the hyperarithmetical hierarchy*, volume 144 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 2000.

[Ash86] C. Ash. Recursive labeling systems and stability of recursive structures in hyperarithmetical degrees. *Trans. Amer. Math. Soc.*, 298:497–514, 1986.

[DKL+15] Rodney G. Downey, Asher M. Kach, Steffen Lempp, Andrew E. M. Lewis-Pye, Antonio Montalbán, and Daniel D. Turetsky. The complexity of computable categoricity. *Adv. Math.*, 268:423–466, 2015.

[DM13] R. Downey and A. Melnikov. Effectively categorical abelian groups. *J. Algebra*, 373:223–248, 2013.

[EG00] Y. Ershov and S. Goncharov. *Constructive models*. Siberian School of Algebra and Logic. Consultants Bureau, New York, 2000.

[Gon80] S. Goncharov. The problem of the number of nonautoequivalent constructivizations. *Algebra i Logika*, 19(6):621–639, 745, 1980.

[Gon81] S. Goncharov. Groups with a finite number of constructivizations. *Dokl. Akad. Nauk SSSR*, 256(2):269–272, 1981.

[Gon97] S. Goncharov. *Countable Boolean algebras and decidability*. Siberian School of Algebra and Logic. Consultants Bureau, New York, 1997.

[Hir01] Denis R. Hirschfeldt. Degree spectra of intrinsically c.e. relations. *J. Symbolic Logic*, 66(2):441–469, 2001.

[LaR77] P. LaRoche. Recursively presented boolean algebras. *Notices AMS*, 24:552–553, 1977.

[LL97] Steffen Lempp and Manuel Lerman. Iterated trees of strategies and priority arguments. *Arch. Math. Logic*, 36(4-5):297–312, 1997. Sacks Symposium (Cambridge, MA, 1993).

[Mal61] A. Mal′cev. Constructive algebras. I. *Uspehi Mat. Nauk*, 16(3 (99)):3–60, 1961.

[Mon13] Antonio Montalbán. A computability theoretic equivalent to Vaught's conjecture. *Adv. Math.*, 235:56–73, 2013.

[Mon14] Antonio Montalbán. Priority arguments via true stages. *J. Symb. Log.*, 79(4):1315–1335, 2014.

[Mon15] Antonio Montalbán. Analytic equivalence relations satisfying hyperarithmetic-is-recursive. *Forum Math. Sigma*, 3:e8, 11, 2015.

[Nur74] A. Nurtazin. *Computable classes and algebraic criteria of autostability*. Summary of Scientific Schools, Math. Inst. SB USSRAS, Novosibirsk, 1974.