EVERY Δ^0_2 DEGREE IS A STRONG DEGREE OF CATEGORICITY

BARBARA F. CSIMA AND KENG MENG NG

Abstract. A strong degree of categoricity is a Turing degree d such that there is a computable structure $\mathcal{A}$ that is d-computably categorical (there is a d-computable isomorphism between any two computable copies of $\mathcal{A}$), and such that there exist two computable copies of $\mathcal{A}$ between which every isomorphism computes d. The question of whether every Δ^0_2 degree is a strong degree of categoricity has been of interest since the first paper on this subject. We answer the question in the affirmative, by constructing an example.

1. Introduction

In Computable Structure Theory, we use Computability Theory to understand the intrinsic complexity of various mathematical objects. One of the most basic notions in mathematics is that of isomorphism. Normally, isomorphism is a notion of sameness. However, there could be two computable copies of the same structure that are not isomorphic via any computable isomorphism. So one can ask the question: What are the possible complexities of isomorphisms of computable structures? That is, which Turing degrees are those that exactly capture the level of difficulty of computing isomorphisms between computable copies of a structure? This question was formalized by Fokina, Kalimullin and R. Miller [FKM10], when they introduced the notion of a *degree of categoricity.*

Definition 1.1 (Fokina, Kalimullin, and Miller [FKM10])**.** A Turing degree $\mathbf{d}$ is said to be the degree of categoricity of a computable structure $\mathcal{A}$ if $\mathbf{d}$ is the least degree such that for every computable copy $\mathcal{B}$ of $\mathcal{A}$, there is a $\mathbf{d}$-computable isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

In that first paper on the subject, Fokina, Kalimullin and Miller showed that every c.e. degree is a degree of categoricity, and indeed that every 2-c.e. degree is a degree of categoricity. They showed that the degrees $\mathbf{0^{(n)}}$ and $\mathbf{0}^{(\omega)}$ are also degrees of categoricity. The question of whether or not every Δ^0_2 degree is a degree of categoricity was the first open question stated at the end of that paper, and answering the question in the affirmative is the main result of this paper. Despite the long wait for an answer to this first question, there has developed a large and interseting body of work on notions related to degrees of categoricity. See Franklin's survey paper [Fra17] for a nice overview. Indeed, prior to this paper, it was not even known if every 3-c.e. degree is a degree of categoricity; the strongest negative result being the existence of a Σ^0_2 degree that is not a degree of categoricity as shown by

Date: March 24, 2022.

The first author is supported by an NSERC Discovery grant.

The second author is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RG23/19).

Anderson and Csima in [AC16]. We refer the reader to [BKY16, BKY18, CFS13, Gon11, BM, CS19] for a collection of related work on degrees of categoricity.

Fokina, Kalimullin and Miller also defined a notion of a *strong degree of categoricity.* A degree of categoricity $\mathbf{d}$ is said to be a strong degree of categoricity if there are particular computable copies $\mathcal{A}$ and $\mathcal{B}$ of a structure with degree of categoricity $\mathbf{d}$ such that every isomorphism between $\mathcal{A}$ and $\mathcal{B}$ computes $\mathbf{d}$. Fokina, Kalimullin and Miller noted that all their examples of degrees of categoricity are actually strong degrees of categoricity, and raised the question of whether all degrees of categoricity are strong; this question remains open.

We prove the following:

Theorem 1.2. *Every Δ^0_2 degree is a strong degree of categoricity.*

Before we begin, we discuss the difficulty encountered trying to extend the ideas from known constructions for structures with 2-c.e. degree of categoricity. Given an approximation of a set D of degree $\mathbf{d}$, the general idea is to build the structure with components for each n in a way that the behavior of the isomorphism on the associated component codes whether or not $n \in D$. In the 2-c.e. coding for a given n, one begins with two distinct nodes that look different. If the first c.e. event happens, one grows each distinct node to still be different, but changes the unique isomorphism. If the second c.e. event happens, one extends to make them look the same; a process we call "homogenizing". Trying to extend beyond 2-c.e., if we try to introduce more coding locations in the same way, we end up losing the D-categoricity of the structure.

To overcome this difficulty, when we code whether a particular $n \in D$, we introduce new coding locations every time there is a change in the approximation. By consulting an isomorphism on a given coding location between our distinct copies, we learn either that the approximation to D on n is correct at the stage, or that there is a further coding location introduced. This is basically like in the original strategy. In order to make sure our structure is D-categorical, we keep track of all possible computable copies of our structure and build D-computable isomorphisms between them. This uses an infinite injury priority argument. The main trick in the basic module, is that when we "de-homogenize" a coding location, we have to be able to correctly recover a D-computable isomorphism. So we make infinite chains of nodes that look distinct stage-by-stage to control recovery, but such that they are isomorphic in the limit. Herein lies the novelty of our method; previous strategies for c.e. and 2-c.e. have focussed on making each coding location in the pair of constructed structures isomorphic at every finite stage. This idea cannot be extended to 3-c.e. as mentioned above, since once we homogenize the coding locations upon observing the second c.e. event, even if we later "de-homogenize" the coding location, we will lose the D-categoricity of the structure. To allow D to be able to differentiate between being in a homogenized and a de-homogenized state, we will potentially grow each coding location infinitely so that isomorphism between the two structures is only achieved in the limit.

2. Requirements

2.1. **Overview.** Fix a computable approximation $\{D_s\}$ of a Δ^0_2 set D. Also fix $\mathcal{B}_i$ to be the i^{th} computable structure of the language. The language will contain infinitely many unary predicates (used to colour the different modules) and graph edges of different colours. We shall build computable structures $\mathcal{A} \cong \mathcal{A}^*$ satisfying that for any isomorphism $f : \mathcal{A} \mapsto \mathcal{A}^*$, we have $f \geq_T D$, and the requirements:

N_i : If $\mathcal{B}_i \cong \mathcal{A}$ then D computes an isomorphism between $\mathcal{B}_i$ and $\mathcal{A}$.

The structure $\mathcal{A}$ will contain infinitely many modules. Different modules are distinguished by the unary predicates. Each module is a collection of distinct graph components called *nodes*. Each node (or graph component) in the module is of a certain type. We will use α, β to refer to nodes in $\mathcal{A}$ (and sometimes $\mathcal{A}^*$) and the letters σ, τ to refer to the (isomorphic images of) nodes in $\mathcal{B}_i$. We use η, ν, ρ, δ to refer to requirements on the construction tree. Each node will consist of a distinguished root vertex r with disjoint chains growing from the root vertex. Each chain is a collection of vertices $r\ v_1\ v_2\ \cdots$ and the vertices are connected by edges of a certain colour k. The k-length of a node is the number of vertices (currently) in the chain with edge colour k. This is like a tree, where every branching is unary, except for the root. See Figure 1.

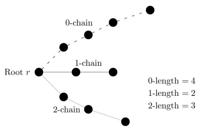

FIGURE 1. A typical node

Now associated with every number $k \geq 0$ we will have a k-chain. At the end each node will have some infinite k-lengths. In each module there are two special nodes which we will call μ_L and μ_R (main left and main right node).

2.2. **Global requirement: Ensuring that any isomorphism $f : \mathcal{A} \to \mathcal{A}^*$ computes D.** A module assigned the coding of $D(n)$ will work in the following way. At the beginning $D(n) = 0$. The module begins by placing μ_L and μ_R in structures $\mathcal{A}$ and $\mathcal{A}^*$. We make μ_L have 0-length 2 and μ_R have 0-length 1 (in both structures). The component will contain nothing else. Hence if this is the final situation then any isomorphism f must map $\mu_L^{\mathcal{A}}$ to $\mu_L^{\mathcal{A}^*}$.

Later on if we ever see n enter D (for the first time) we will switch by growing $\mu_L^{\mathcal{A}}$ and $\mu_R^{\mathcal{A}^*}$ to have 0-length 2 and $\mu_R^{\mathcal{A}}$ and $\mu_L^{\mathcal{A}^*}$ to have 0-length 3. At the same time we begin a new module (in some unused part of the structure) and repeat this entire process, but with $D(n) = 0$ replaced with $D(n) = 1$.

If n is later extracted from D we will make $\mu_L \cong \mu_R$ (in the original module) by growing their 0-lengths to 3. Note: the homogenization at this step is not necessary for coding D into f, but will be necessary to ensure meet the N_i requirements. Later on if n re-enters D we will switch again (in the original module) by making $\mu_L^{\mathcal{A}}$ and $\mu_R^{\mathcal{A}^*}$ have 0-length 3 and $\mu_R^{\mathcal{A}}$ and $\mu_L^{\mathcal{A}^*}$ to have 0-length 4. And so on.

Now we check that this allows any isomorphism $f : \mathcal{A} \mapsto \mathcal{A}^*$ to compute D. To figure out if $n \in D$ we begin by considering the initial/original module (in $\mathcal{A}$ and $\mathcal{A}^*$) which codes n. We compute $f\left(\mu_L^{\mathcal{A}}\right)$. If this is equal to $\mu_L^{\mathcal{A}^*}$ then we conclude that $D(n) = 0$, because by the construction, if $D(n) = 1$ we always switch. However if we find that $f\left(\mu_L^{\mathcal{A}}\right) = \mu_R^{\mathcal{A}^*}$ then we will know that there is some least stage s where n enters D at stage s. Find this stage s and locate the new module which is started at stage s. Inductively, apply f on this new module. The fact that D

is Δ^0_2 means that only finitely many new modules are started and this process will terminate after finitely many steps.

So now we must describe how to implement this global strategy and still satisfy all requirements N_i. Now for the rest of this discussion we will fix n and a particular module assigned to code n. The construction will be uniform with respect to the modules. In order for the global strategy to work, in this module, we must ensure that:

- if $D(n) = 0$ with no changes, then $\mu^A_L \cong \mu^{A^*}_L$ and $\mu^A_R \cong \mu^{A^*}_R$ but $\mu^A_L \ncong \mu^A_R$. These are the only nodes in the module. Hence any f must map μ^A_L to $\mu^{A^*}_L$.
- if $D(n) = 1$ then $\mu^A_L \ncong \mu^{A^*}_L$.

As long as these two are satisfied, the global strategy will work, as we only apply f on μ^A_L. In particular, once n is enumerated in D for the first time (if ever), we are allowed to add other nodes into the module, and the global strategy will not care about what we do to these nodes. We will now describe what are the extra nodes added, and assume, for notational simplicity that we add these extra nodes in $\mathcal{A}$. Of course we will mirror these actions in $\mathcal{A}^*$ but we will suppress mention of this. In other words, once n is enumerated in D we will begin adding and growing new nodes in $\mathcal{A}$ and duplicate these actions in $\mathcal{A}^*$. We will also occasionally grow μ^A_L and duplicate these actions for $\mu^{A^*}_R$. Similarly when we grow μ^A_R we will duplicate this for $\mu^{A^*}_L$.

To satisfy N_i we will describe, uniformly, how to map nodes in this module of A to the corresponding nodes in $\mathcal{B}_i$. Since we can computably distinguish between different components (using the unary predicates), this uniformity will allow us to construct a computable (in D) isomorphism between $\mathcal{A}$ and $\mathcal{B}_i$.

2.3. **Notations used.** We shall require the following parameters to organize the construction:

- For each $\eta \in 2^{i-1}$, we shall define a partial computable functional g_η, so that if η is an initial segment of the true path, g^D_η will be an isomorphism from $\mathcal{A}$ to $\mathcal{B}_i$. Each module of $\mathcal{A}$ will be working towards coding membership of some n in D, so it will be sufficient for the map g^D_η to only consult the oracle about membership of n in D for any node in that module. To ease notation, when we are restricting our attention to a module coding membership of n in D, we let g^0_η denote the map that will be correct if the approximation to n in D does not change after the node α was introduced, and g^1_η denote the map that will be correct otherwise. If α is a node in a module coding n, we will let $f_\eta(\alpha)$ be the first node in $\mathcal{B}_i$ that appears isomorphic to α at an η-recovery stage, and let $g^0_\eta(\alpha) = f_\eta(\alpha)$. Though $f_\eta = g^0_\eta$ here, the notion of the "first node" map will be helpful to us. We will define $g^1_\eta(\alpha)$ more cleverly, so that α and $g^1_\eta(\alpha)$ will be isomorphic if the approximation to n in D changes and η is on the true path. Note, however, that α and $g^1_\eta(\alpha)$ will *not* be isomorphic at any stage of the construction.
- There are two operations which generate new nodes in $\mathcal{A}$: a *forward* operation $\mathbb{F}_\eta$ and a *backward* operation $\mathbb{B}_\eta$. Nodes $\alpha \in \mathcal{A}$ will require us to define $g^1_\eta(\alpha)$. For certain reasons we do not wish to define $g^1_\eta(\alpha)$ to be some node already present in $\mathcal{B}_{|\eta|}$. The operation $\mathbb{F}_\eta$ will be used to generate a new node $\mathbb{F}_\eta(\alpha) \in \mathcal{A}$ which will force $\mathcal{B}_{|\eta|}$ to produce a new node which can then be used as $g^1_\eta(\alpha)$. That is, we will define $g^1_\eta(\alpha) = f_\eta(\mathbb{F}_\eta(\alpha))$. The

idea is that by controlling the growth of $\mathbb{F}_\eta(\alpha)$ we can restrain $g^1_\eta(\alpha)$ from growing unexpectedly and hence ensure $\alpha \cong g^1_\eta(\alpha)$ in the limit. Of course the newly introduced node $\mathbb{F}_\eta(\alpha)$ will now require us to define $g^1_\eta(\mathbb{F}_\eta(\alpha))$; for this reason we will need to introduce an infinite forward sequence α, $\mathbb{F}_\eta(\alpha)$, $\mathbb{F}^2_\eta(\alpha), \dots, \mathbb{F}^k_\eta(\alpha), \dots$

In order to ensure that the map g^1_η is onto, we will also need an infinite backward sequence $\mathbb{B}_\eta(\alpha)$, $\mathbb{B}^2_\eta(\alpha), \dots \mathbb{B}^1_\eta(\alpha), \dots \mathbb{B}^n_\eta(\alpha), \dots$.

We will often refer to such nodes introduced into $\mathcal{A}$ for the sake of the requirement η as *η-nodes.*

3. One requirement N_i in isolation

3.1. **Initial $n \notin D$.** We now consider one requirement in isolation, and fix a particular module coding n. Suppose N_i is assigned to η of the priority tree (right now we have not yet discussed the priority tree). Let m be a unique number associated to this module and guess η. At the beginning before n enters D we have no choice but to only have the main nodes μ_L and μ_R with 0-lengths 2 and 1, respectively. We wait for $\mathcal{B}_i$ to respond with nodes that we can use to define $g^0_\eta(\mu_L)$ and $g^0_\eta(\mu_R)$. Nothing else interesting happens at this point.

3.2. n **enters** D**.** Now assume that at some stage n first enters D. We switch the main nodes in $\mathcal{A}$ and $\mathcal{A}^*$ as described in Section 2.2. So now, in $\mathcal{A}$, μ_L has 0-length 2 and μ_R has 0-length 3. Wait for $\mathcal{B}_i$ to catch up. Note that now we may have $g^0_\eta(\mu_L) \cong \mu_R$ and vice-versa. Now we need to grow, step by step, the following sequences:

$$\cdots \to \mathbb{B}^2_\eta(\mu_L) \to \mathbb{B}_\eta(\mu_L) \to \mu_L \to \mathbb{F}_\eta(\mu_L) \to \mathbb{F}^2_\eta(\mu_L) \to \cdots$$

$$\cdots \to \mathbb{B}^2_\eta(\mu_R) \to \mathbb{B}_\eta(\mu_R) \to \mu_R \to \mathbb{F}_\eta(\mu_R) \to \mathbb{F}^2_\eta(\mu_R) \to \cdots$$

Since this is the basic action in the construction, we now describe how these sequences are formed and will help us define g^1_η correctly. At the start we only have the nodes μ_L and μ_R in $\mathcal{A}$ and $g^0_\eta(\mu_L)$ and $g^0_\eta(\mu_R)$ in $\mathcal{B}_{|\eta|}$.

First we grow a new node in $\mathcal{A}$ which we label with $\mathbb{F}_\eta(\mu_L)$. We first grow this new node to have m-length 2, introduce no other chains, and wait for $\mathcal{B}_i$ to respond. Since the existing nodes in $\mathcal{B}_i$ all have non-trivial 0-lengths, the only way to catch up is for $\mathcal{B}_i$ to grow a new node to have m-length 2. Once we see this we then grow $\mathbb{F}_\eta(\mu_L)$ to have 0-length 2. Again the only way for $\mathcal{B}_i$ to catch up is to respond by growing the same node σ. Once we see this, we define $g^1_\eta(\mu_L) = \sigma$ and define $g^0_\eta(\mathbb{F}_\eta(\mu_L)) = \sigma$. This completes one round of the basic action.

Now at the next round we grow a new node in $\mathcal{A}$ labeled $\mathbb{B}_\eta(\mu_L)$. We would like $\mathbb{B}_\eta(\mu_L) \cong g^0_\eta(\mu_L)$ which currently has 0-length 2 or 3. Let's assume that $g^0_\eta(\mu_L)$ currently has 0-length 3. So we grow $\mathbb{B}_\eta(\mu_L)$ to have the same 0-length 3. We again do this in a two step process as above: First, grow $\mathbb{B}_\eta(\mu_L)$ to have m-length 4 and no other chains. When $\mathcal{B}_i$ responds with a new node, say σ', we grow $\mathbb{B}_\eta(\mu_L)$ to have 0-length 3. Again $\mathcal{B}_i$ must respond by growing the same node. We then define $g^1_\eta(\mathbb{B}_\eta(\mu_L)) = f_\eta(\mu_L)$ and $g^0_\eta(\mathbb{B}_\eta(\mu_L)) = \sigma'$.

For the third and fourth rounds repeat symmetrically by growing $\mathbb{F}_\eta(\mu_R)$ and $\mathbb{B}_\eta(\mu_R)$ in $\mathcal{A}$, making sure that each node is introduced with an m-length two greater than the one before. At the end of the fourth round there are six nodes in each structure, with all the non-main nodes having distinct m-lengths that are off by at least two from one another. See Figure 2.

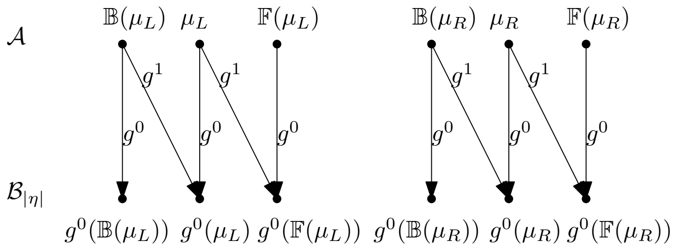

FIGURE 2. End of the fourth round.

Now observe that we still need to define g^1_η on $\mathbb{F}_\eta(\mu_L)$ and $\mathbb{F}_\eta(\mu_R)$, and we need to define $(g^1_\eta)^{-1}$ on two nodes in $\mathcal{B}_i$. After completing these four rounds we say that we have *completed a single η-cycle.*

At the end of a cycle we grow m-lengths. Increase every m-length in every node by one. Since all distinct m-lengths are off by two from one another, this will ensure that $f_\eta(\sigma) \cong \sigma$ for each η-node σ at each η-recovery stage.

Now we begin the next cycle and define $\mathbb{F}^2_\eta(\mu_L), \mathbb{F}^2_\eta(\mu_R), \mathbb{B}^2_\eta(\mu_L)$ and $\mathbb{B}^2_\eta(\mu_R)$. We also extend g^0_η and g^1_η. At the end of each cycle we always grow the m-lengths.

Figure 3 depicts the structure after four cycles.

If $D(n) = 1$ and $\mathcal{B}_i \cong \mathcal{A}$ then we complete infinitely many cycles.

If $D(n) = 1$ forever with no extraction, then it is easy to see that g^1_η is a bijection. Every node will eventually have m-length ∞, and the 0-lengths of all nodes remain as they were at the stage when n first entered D. Hence $\mu_L \not\cong \mu_R$ (as required by the global strategy) and $\alpha \cong g^1_\eta(\alpha)$ for every $\alpha \in \mathcal{A}$.

3.3. n **leaves** D**.** Let's consider the case when n leaves D after finitely many cycles. We need to homogenize the module, because currently $\mu_L \not\cong g^0_\eta(\mu_L)$. Assume that the 0-lengths are x and $x+1$. We will now make μ_L and μ_R have the same 0-length $x+1$. We grow all $\mathcal{A}$-nodes accordingly and *stop putting new nodes* into this module of $\mathcal{A}$. Wait for $\mathcal{B}_i$ to catch up. It is again easy to check that every $\mathcal{B}_i$-node must catch up accordingly, i.e. $f_\eta(\sigma) \cong \sigma$, because of the uniqueness of the m-lengths of the non-main nodes, and the fact that the main nodes are now isomorphic. Let's again call this completing *a single η-cycle.*

For the next cycle we increase all the m-lengths all by one.

Now if n never re-enters D, the effect of completing infinitely many such η-cycles is to make every node in $\mathcal{A}$ and $\mathcal{B}_i$ have 0-length $x+1$, and every non-trivial m-length to be ∞. This means that the module is completely homogeneous and so g^0_η

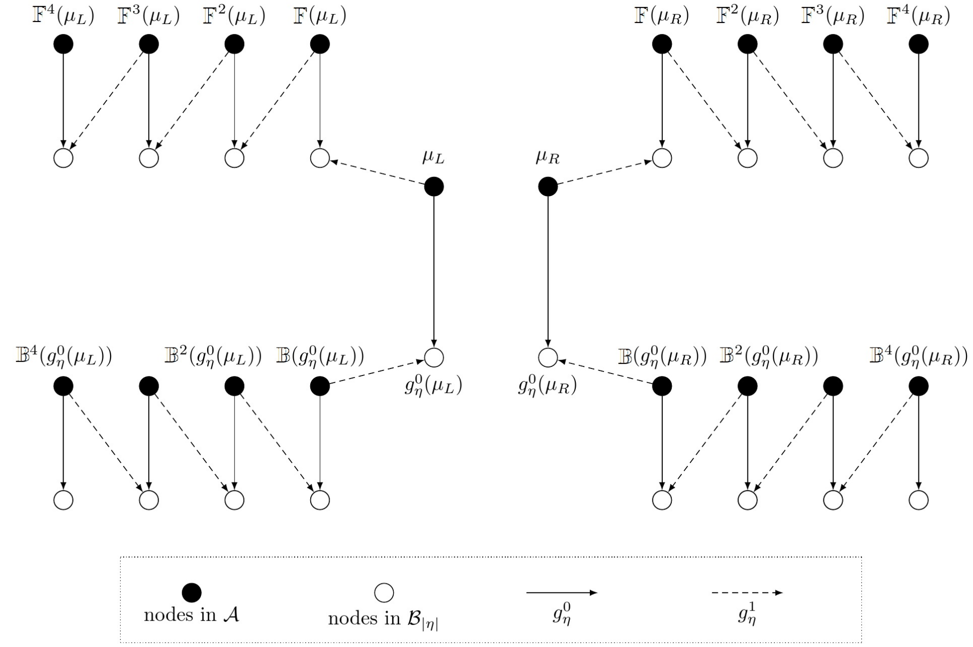

FIGURE 3. After 4 cycles

is an isomorphism of the module. In this case we only have finitely many nodes as we never introduce a new node when $n \notin D$. Also the global requirement does not care what we do in this module as $D(n) = 0$ and n enumerated at least once in D.

3.4. n **re-enters** D**.** Now assume that at some point after finitely many η-cycles (while $D(n) = 0$), we find that n is re-enumerated in D. Assume that at this point the 0-length of μ_L and μ_R is x. Now we are going to perform what we call *η-preparation.* This consists of the following steps.

3.4.1. *Separating* μ_L, μ_R. Notice that with the exception of μ_L and μ_R which are identical to each other, all nodes in $\mathcal{A}$ are distinct due to their unique m-lengths. Since n has entered D, we must again distinguish μ_L and μ_R. We *first* grow the 0-types of the main nodes, i.e. keep μ_L with 0-length x and grow μ_R to have 0-length $x + 1$. Leave all other nodes for the time being. Wait for $\mathcal{B}_i$ to catch up (if we were in the middle of a cycle, we also complete the cycle). At this point, either $\mu_L \cong g_\eta^0(\mu_L)$ and $\mu_R \cong g_\eta^0(\mu_R)$, or $\mu_L \cong g_\eta^0(\mu_R)$ and $\mu_R \cong g_\eta^0(\mu_L)$. In the first case, increase the 0-length of all the forward and backward nodes of μ_R by one. In the second case, increase the 0-length of all forward nodes of μ_R by one, and all backward nodes of μ_L by one (the "wrong" recovery has caused us to "flip" the backward sequences).

3.4.2. *Resuming the construction of the* $\mathbb{F}$ *and* $\mathbb{B}$ *sequences.* We now continue the process of introducing more forward and backward nodes in our sequences for μ_L and μ_R. We introduce the nodes so that they have longer and distinct m-lengths that are off by two from one another, and such that the 0-length of $\mathbb{F}(\alpha)$ agrees with the 0-length of α, and also the 0-length of $\mathbb{B}(\alpha)$ agrees with the 0-length of α.

We note that at the end, if there are infinitely many η-expansionary stages, then for every $k \neq 0$, every k-length of every node in $\mathcal{A}, \mathcal{B}_i$ will be ∞. So in the limit, the nodes are only differentiated by their 0-length if $D(n) = 1$, and are indistinguishable if $D(n) = 0$.

4. Two requirements, $\nu = \eta * fin$

We now consider two strategies, η and ν. Here ν is of lower priority, working on meeting requirement N_j, and believes in the finitary outcome of η, which is working on meeting requirement N_i. The outcomes of η are the usual ones, guessing whether $\mathcal{B}_i \cong \mathcal{A}$.

Roughly speaking, η and ν pursue their strategies as described in Section 3, with very little interaction. We now describe how this will work in a given module. The module will still have the nodes μ_L and μ_R, and the 0-lengths of these nodes will be dictated by the approximation to D on the relevant value for this module, exactly as in Section 3. But now, η has to define an isomorphism between $\mathcal{A}$ and $\mathcal{B}_i$, while ν has to define an isomorphism between $\mathcal{A}$ and $\mathcal{B}_j$. So they will both be putting nodes into $\mathcal{A}$, and they must correctly map nodes that they did not introduce. Let m_η and m_ν be unique numbers associated to this module and the guesses η and ν, respectively. Since ν believes the finitary recovery of η, its strategy will be reset at each η-recovery stage. When this happens, m_ν will be reset to a new, unused value.

The strategy for η proceeds exactly as described in Section 3, building $\mathbb{F}_\eta$ and $\mathbb{B}_\eta$ sequences for μ_L and μ_R, using the number m_η for m. At some point, we are ready to consider ν. Since ν is guessing the finitary recovery for η, it assumes that there will be no more η-recovery stages, and so η will not be introducing any further nodes into $\mathcal{A}$. The ν strategy considers the finitely many $\mathbb{F}_\eta$ and $\mathbb{B}_\eta$ nodes that have already been introduced to be garbage. Note that all these η-nodes have distinct m_η-lengths. At the next ν-recovery, when $\mathcal{B}_j$ exhibits the unique nodes, the ν strategy sets both g^0_ν and g^1_ν to map the η-nodes to their matching nodes in $\mathcal{B}_j$. That is, $g^0_\eta = g^1_\eta = f_\eta$ on these nodes. Now ν proceeds as in Section 3, building $\mathbb{F}_\eta$ and $\mathbb{B}_\eta$ sequences for μ_L and μ_R, using the number m_ν for m. The only modification is that at the point of the cycle when we extend the 0-length of a new ν-nodes, we also extend the m_η-length to match the m_η-lengths in μ_L and μ_R. So if there are indeed no more η-recovery stages, then the strategy for ν will work out almost exactly as in Section 3, with just a few extra nodes floating about with distinguished m_η-lengths, and the shortest m_η length found in μ_L and μ_R repeated in all other nodes. Now if at some point the η-strategy *does* recover, it will similarly view all the ν-nodes as garbage, and both g^0_η and g^1_η will use f_η to map them at the next η-recovery. All m_ν-lengths will be increased to be equal to one another. The η strategy will continue building its $\mathbb{F}_\eta$ and $\mathbb{B}_\eta$ sequences, but at the point of copying the 0-length will also copy the final m_ν-length. When μ restarts, it will use a new value for m_ν, and proceed as before.

5. Two requirements, $\nu = \eta * \infty$

We now consider two strategies, η and ν, where ν is of lower priority, working on N_j, and believes in the infinitary outcome of η, which is working on N_i. So ν only gets to act at $\eta * \infty$ stages.

Let us again consider what happens in a given module. As before, let m_η and m_ν be unique numbers associated to this module and the guesses η and ν, respectively. Since η has highest priority, it must be allowed to proceed with its own strategy without waiting for ν, which it does as described in the earlier sections. Since ν believes in the ∞ outcome of η, it must decide where to map the nodes η introduces. It cannot ask for η to wait, and $\mathcal{B}_j$ might be slower to recover than $\mathcal{B}_i$. So even though the η-nodes have distinct m_η-lengths, this does not help ν with defining an isomorphism between $\mathcal{A}$ and $\mathcal{B}_j$. The solution is for ν to make separate $\mathbb{F}_\nu$ and $\mathbb{B}_\nu$ nodes for *each* of the η-nodes. All the different ν-nodes will have unique m_ν-lengths off by at least two from one another at each stage, so ν can work with the same number m_ν in all its nodes. As the m_ν-length grows in μ_L and μ_R, this will be reflected back to the η-nodes at η-recovery stages (at the point where the 0-lengths were caught up in the basic strategy.)

6. The formal construction

We now give the formal construction, following the plan outlined in the previous sections.

Let $\mathbf{d} \leq \mathbf{0}'$ be arbitrary, and fix $D \in \mathbf{d}$ with computable approximation $\{D_s\}_{s\in\omega}$. Adopt the convention that $D_s(n) = 0$ for all $s \leq n$. We will construct a structure with degree of categoricity $\mathbf{d}$. The structure will be over the language $\mathcal{L} = \{U_{n,k}, E_n\}_{n,k\in\omega}$. The $U_{n,k}$ will be unary relations (that will partition the universe). The E_n will be binary relations (thought of as edges with colour n). Note that we have chosen an infinite computable language purely out of convenience. If we wanted a finite language, we could simply simulate the predicates $U_{n,k}$ and E_n by attaching cycles of different lengths.

We will build computable copies $\mathcal{A}$ and $\mathcal{A}^*$ of the same structure, such that $\mathcal{A}$ has strong degree of categoricity $\mathbf{d}$ as witnessed by $\mathcal{A}$ and $\mathcal{A}^*$. We will build $\mathcal{A}$, and we will specify how what we build in $\mathcal{A}$ should be reflected into $\mathcal{A}^*$. The structures $\mathcal{A}$ and $\mathcal{A}^*$ will be isomorphic at each stage of the construction, as well as in the limit.

The purpose of the $U_{n,k}$ is to separate the modules for coding membership into D. Each member of $\mathcal{A}$ will have exactly one of the $U_{n,k}$ hold of it, and the members of $U_{n,k}$ will be the module coding whether the approximation to $D(n)$ changes a $k+1$st time. In particular $U^{\mathcal{A}}_{n,k} \neq \emptyset \Leftrightarrow |\{s \mid D_{s+1}(n) \neq D_s(n)\}| \geq k$.

We must define functionals g_η for $\eta \in 2^{<\omega}$ such that if η is on the true path of our construction and $\mathcal{B}_{|\eta|} \cong \mathcal{A}$ then $g^D_\eta : \mathcal{A} \cong \mathcal{B}_{|\eta|}$. For $\alpha \in \mathcal{A}$, first check which module α belongs to. If $\alpha \in U_{n,k}$, then this means the approximation to n in D changed at least k times. Let s denote the stage when the first member was introduced into $U_{n,k}$, or equivalently the stage when the approximation to n in D changed for the kth time. Then we let $g^D_\eta(\alpha) = g^0_\eta(\alpha)$ if $D(n) = D_s(n)$, and $g^D_\eta(\alpha) = g^1_\eta(\alpha)$ otherwise, where g^0_η and g^1_η will be defined during the construction. We will also define maps $f_\eta : \mathcal{A} \to \mathcal{B}_{|\eta|}$ which at η-recovery stages will be injective maps that take nodes in $\mathcal{A}$ to nodes in $\mathcal{B}_{|\eta|}$ in such a way that important m-lengths are preserved.

All nodes in the structures will have a node type of either "basic", or "ρ" for some $\rho \in 2^{<\omega}$. As such, nodes will be referred to as "basic nodes" or "ρ nodes". A node's type will be specified when it is introduced, and may change during the construction.

Definition 6.1. We say a map $h : \mathcal{A}[t] \to \mathcal{B}_{|\eta|}[s]$ from nodes of $\mathcal{A}[t]$ to nodes of $\mathcal{B}_{|\eta|}[s]$ is *length preserving for* η if for each $\alpha \in A[t]$, $h(\alpha)$ is a node in $\mathcal{B}_{|\eta|}[s]$ with the same m-lengths as α except for $m = 0$ or $m = m_\rho[s]$ with $\rho^\frown 0 \preceq \eta$ or $\rho \geq \eta^\frown 1$.

We now define what it means for s to be an η-stage.

Stage 0: 0 is an η-stage for all $\eta \in 2^{<\omega}$.

Stage $s > 0$: s is a λ-stage. Suppose s is an η-stage for some $\eta \in 2^{<\omega}$ with $|\eta| < s$. Let $t < s$ be the greatest such that t is a τ-stage for some $\tau \leq \eta^\frown 0$ with $|\tau| = |\eta^\frown 0|$. We say s is an $\eta^\frown 0$-stage, and also say s is an η*-recovery stage* if η is active, and there is an injective map $h : \mathcal{A}[t] \to \mathcal{B}_{|\eta|}[s]$ from nodes of $\mathcal{A}[t]$ to nodes of $\mathcal{B}_{|\eta|}[s]$ that has the following properties:

- h is length preserving for η
- $h \supseteq f_\eta[t]$
- if $\mathrm{dom}(f_\eta[t]) \neq \mathcal{A}[t]$ and $\mathrm{rng}(f_\eta[t]) \neq \mathcal{B}[s]$ then the least node $\beta \in \mathcal{B}_{|\eta|}[t]$ with $\beta \notin \mathrm{rng}(f_\eta[t])$ belongs to the range of h
- for basic nodes $\alpha \in \mathcal{A}[t]$, the 0-length of $h(\alpha)$ agrees with the 0-length of $\alpha[s-1]$

If s is an η-recovery stage, we also define $s_\eta = t$. If s is not an η-recovery stage, we say s is an $\eta^\frown 1$-stage. Let $\eta_s \in 2^s$ be such that s is an η_s-stage. (Note that η_s exists and is unique.)

Construction. Stage 0: Introduce basic nodes μ_L with 0-length 2 and μ_R with 0-length 1 into $U_{0,0}$. Declare all strategies to be active.

Stage $s > 0$:

We first make sure all nodes of the form μ_L and μ_R are correctly coding membership in D.

For $n = s$ and each n such that $D_s(n) \neq D_{s-1}(n)$, do as follows. Let k be least such that $U_{n,k} = \emptyset$. Introduce basic nodes $\mu_L^{\mathcal{A}}$ and $\mu_L^{\mathcal{A}^*}$ (with 0-length 2), and $\mu_R^{\mathcal{A}}$ and $\mu_R^{\mathcal{A}^*}$ (with 0-length 1) into $U_{n,k}$.

If $k \geq 1$, then in $U_{n,k-1}$, grow $\mu_R^{\mathcal{A}^*}$ to have 0-length 2, and $\mu_R^{\mathcal{A}}$ and $\mu_L^{\mathcal{A}^*}$ to have 0-length 3. Make the commitment that henceforth $\mu_L^{\mathcal{A}^*} \cong \mu_R^{\mathcal{A}}$ and $\mu_R^{\mathcal{A}^*} \cong \mu_L^{\mathcal{A}}$. That is, if ever we say to grow μ_L in a certain way, we really mean both $\mu_L^{\mathcal{A}}$ and $\mu_R^{\mathcal{A}^*}$, and symmetrically with L and R switched.

If $k \geq 2$, then for all $l \leq k-2$ do as follows. In $U_{n,l}$, if the 0-length of μ_L and the 0-length of μ_R are the same, increase the 0-length of μ_R by one. If the 0-length of μ_R is one greater than the 0-length of μ_L, extend the 0-lengths of all nodes in $U_{n,l}$ to have 0-length equal to that of μ_R.

Now we act for various strategies η based on our current approximation η_s.

For all $\nu > \eta_s$, undefine f_ν, g^0_ν and g^1_ν on any values they have been defined on, and reset the ν strategy as follows. Increase all m_ν-lengths to the maximum current m_ν length. As will be immediate from their forthcoming definitions, each ν-node β is of the form $\mathbb{F}^l_\nu(\alpha)$ or $\mathbb{B}^l_\nu(\alpha)$ for some $l \geq 1$ and some node α that is either a basic node or ρ-node for some ρ with $\rho \preceq \nu$. Now that the m_ν-lengths of α and β are equal, we have that α is isomorphic to β. Make the commitment that henceforth the node β copies α, and that the label and node type of β is the same as the label and node type of α. We cancel all ν-cycles, re-set m_ν to a new value, and declare ν to be active.

If there is an η with $\eta \preceq \eta_s$ such that $\mathcal{B}_{|\eta|}[s]$ contains extra copies or proper extensions of nodes with non-minimal m_ρ-length for any ρ such that $\rho \leq \eta$ and $\rho\widehat{\ }0 \not\preceq \eta$, or ρ such that $\rho \succeq \eta\widehat{\ }0$ as compared to $\mathcal{A}[s-1]$, we choose the highest priority such η and act to meet $N_{|\eta|}$ by ensuring $\mathcal{B}_{|\eta|} \ncong \mathcal{A}$. Declare η to be inactive. Proceed to the next stage of the construction.

Otherwise, proceed as follows.

For all η with $\eta\widehat{\ }0 \preceq \eta_s$ (initial segments of η_s for which we have recovery), do as follows.

For each node α that was present at stage s_η and on which it is not already defined, we define $f_\eta(\alpha) = h(\alpha)$, where h is the map witnessing that s is an η-recovery stage. We will have $g_\eta^0 = f_\eta$ on all nodes where f_η is defined.

For all ρ-nodes α with $\rho\widehat{\ }0 \npreceq \eta\widehat{\ }0$, if $f_\eta(\alpha)$ is defined but g_η^1 is not already defined, we define $g_\eta^1(\alpha) = f_\eta(\alpha)$.

For each n, k such that $U_{n,k+1} \neq \emptyset$, do as follows in $U_{n,k}$. If η is active and there is an active η-cycle, perform the next step as prescribed by that cycle. If there is not an active η-cycle, we activate one. At each η-stage, one step is taken in the η-cycle. As we are now activating the η-cycle, we describe the steps that will need to be taken to complete it.

Step 1: For each node α that is either basic, or a ρ-node with $\rho\widehat{\ }0 \preceq \eta$, and on which $f_\eta(\alpha)$ has been defined, for the least l for which $\mathbb{F}_\eta^l(\alpha)$ and $\mathbb{B}_\eta^l(\alpha)$ have not been defined, we build $\mathbb{F}_\eta^l(\alpha)$ and $\mathbb{B}_\eta^l(\alpha)$ in $\mathcal{A}$ during this step. List the finitely many nodes to be built during this step, and assign them distinct m_η-lengths, all greater than the m_η-length of μ_L, and such that all the lengths have a difference of at least 2 between each other (so that later we can simultaneously grow them all by one and have them distinct from each other and the lengths at the previous stage). For each of the nodes being created in this step, introduce a new labeled η-node with an m_η-chain of the assigned length, and an m-chain of minimal m-length for all $m \neq m_\eta$ that have a non-trivial m-length in another node of the module.

Step 2: Define the map g_η^1 on the newly introduced nodes. For α a basic node or a ρ-node with $\rho\widehat{\ }0 \preceq \eta$, $g_\eta^1(\mathbb{F}_\eta^{l-1}(\alpha)) = f_\eta(\mathbb{F}_\eta^l(\alpha))$, and $g_\eta^1(\mathbb{B}_\eta^l(\alpha)) = f_\eta(\mathbb{B}_\eta^{l-1}(\alpha))$. (Here $\mathbb{F}_\eta^0(\alpha) = \mathbb{B}_\eta^0(\alpha) = \alpha$.)

The next two steps deal with all η-nodes, not just the new ones.

Step 3: For any η-nodes of the form $\mathbb{F}_\eta^l(\alpha)$, increase all 0-lengths to that of α. For any η-nodes of the form $\mathbb{B}_\eta^l(\alpha)$, increase all 0-lengths to that of $f_\eta(\mathbb{B}_\eta^{l-1}(\alpha))$.

Step 4: Increase the m_η-length of each η-node by one.

Step 5: Let m be the minimum of the m_η-lengths of η-nodes at the end of step 4. For any node α which is *not* an η-node, increase the m_η-length of α to $m-2$. (The idea is that these are nodes α for which η may have started forward and backward sequences. The m_η-length is not being used to distinguish between these nodes, but must tend to infinity if the relevant map is to be correct in the limit. We want the m_η-length of non η-nodes to be equal.)

This completes the instructions for what must happen in a single η-cycle. That is, once step 5 is complete, we say that the η-cycle is complete, and there is no active η-cycle until one is activated during the next η-recovery stage.

This completes the construction.

Verification. We begin with some observations about the construction.

At a given stage, if α is an η-node, then for $m \neq m_\eta, 0$, the m-length of η is minimal. The m_η-length of α agrees only with the m_η-lengths of nodes with the same label as α, and differs by at least two from the m_η-length of all other nodes.

After it is introduced, a node α can only have its label changed finitely often. So there is some $\rho \leq TP$ and some stage s such that it is a ρ-node at all stages beyond s. In this case we will say the node is a ρ-node at the end of the construction, and that its label at the end of the construction is the label it had at stage s.

We note that the nodes in $\mathcal{A}$ at the end of the construction will look as follows. Firstly, each node will have infinite m-lengths for exactly those m that are the final value of m_ρ for some ρ with $\rho\frown 0 \prec TP$. So it remains to describe the m-lengths for all other m, which will all be finite. The 0-lengths will all be finite, and will either all be the same or will be one of two values that differ by one. If m is not m_ρ for any ρ at the end of the construction, then all m-lengths will be equal and minimal. If α is a ρ-node with $\rho \leq TP$ it may have non-minimal m-lengths only for $m = m_\rho$.

Lemma 6.2. *Suppose η is on the true path and $\mathcal{A} \cong \mathcal{B}_{|\eta|}$, and that s_0 is such that $\eta_s \geq \eta$ for all $s \geq s_0$. Then η is active at all stages $s \geq s_0$.*

Proof. Assume for a contradiction that η is declared inactive at some stage $s \geq s_0$. Then $\mathcal{B}_{|\eta|}[s]$ contains an extra copy or proper extension of a node with non-minimal m_ρ-length as compared to $\mathcal{A}[s-1]$ for some ρ such that $\rho \leq \eta$ and $\rho \not\prec \eta$, or ρ such that $\rho \succeq \eta$. We have no η-recovery stages after s since we have declared η inactive at stage s and it cannot be reactivated since $s \geq s_0$. Thus there can be no extension to or additions of nodes with non-trivial m_ρ-length to $\mathcal{A}$ after stage $s-1$, so $\mathcal{B}_{|\eta|} \ncong \mathcal{A}$, a contradiction. Hence η is active at all stages greater than or equal to s_0. □

Lemma 6.3. *Suppose η is on the true path and $\mathcal{A} \cong \mathcal{B}_{|\eta|}$, and that s_0 is such that $\eta_s \geq \eta$ for all $s \geq s_0$. Then for any node α and any $s > s_0$ such that there exists an injective map of nodes $\mathcal{A}[s_\eta] \to \mathcal{B}_{|\eta|}[s]$ that is length preserving for η, if $f_\eta(\alpha)[s]$ is defined then it has the same m-lengths as α except for $m_\rho[s]$ with $\rho\frown 0 \preceq \eta$ or $\rho \geq \eta\frown 1$, or 0. Moreover, if α is not a basic node or a ρ-node with $\rho\frown 0 \prec \eta$ then the 0-length is also the same.*

Proof. We restrict our attention to a fixed module. Suppose $f_\eta(\alpha)$ was defined with the desired properties at stage s_η. We verify that $f_\eta(\alpha)$ has the desired properties at stage s.

If we care about the m-length, and $m \neq m_\eta$, then $m = m_\rho$ with $\rho\frown 1 \preceq \eta$, $\rho < \eta_s$, or $\eta\frown 0 \preceq \rho$. In all these cases the m-length does not change between the end of stage s_η and the beginning of stage s.

We consider different cases depending on the node type of α.

If α is an η-node, then the m_η-length of α at stage s_η agrees only with the m_η-lengths of nodes with the same label as α, and differs by two from the m_η-length of all other nodes. Depending on which step we are in at stage s_η, the m_η-lengths of all η-nodes present at stage s_η either remain unchanged or increase by one. There is no change to m_η-lengths between the stages s_η and s. Nodes with the same label agree on all m-lengths at all times. Since η is active at stage s, $\mathcal{B}_{|\eta|}[s]$ does not have extra copies of nodes with non-minimal m_η-length as compared to $\mathcal{A}[s-1]$. So since we have an injection that is length preserving for η from $\mathcal{A}[s_\eta]$ to $\mathcal{B}_{|\eta|}[s]$, the node $f_\eta(\alpha)[s]$ must have grown into (or remained) a node with the required properties.

If α is a ρ-node with $\rho < \eta\widehat{\ }0$, $\rho \not\preceq \eta$, then the m_ρ-length of α at stage s_η agrees only with the m_ρ-lengths of nodes with the same label as α, and differs by two from the m_ρ-length of all other nodes. As ρ is to the left of the true path, there is no change to m_ρ-lengths at any further point in the construction beyond s_0, and there are only the finitely many ρ-nodes in $\mathcal{A}$ that were present in $\mathcal{A}[s_0]$. Since $\mathcal{B}_{|\eta|} \cong \mathcal{A}$, there cannot be new nodes with non-minimal m_ρ-length in $\mathcal{B}_{|\eta|}[s]$ as compared to $\mathcal{B}_{|\eta|}[s_\eta]$. Nodes with the same label agree on all m-lengths at all times. So since we have an injection that is length preserving for η from $\mathcal{A}[s_\eta]$ to $\mathcal{B}_{|\eta|}[s]$, the node $f_\eta(\alpha)[s]$ must have grown into (or remained) a node with the required properties.

If α is a ρ-node with $\rho\widehat{\ }1 \preceq \eta$, then the m_ρ-length of α at stage s_η agrees only with the m_ρ-lengths of nodes with the same label as α, and differs by two from the m_ρ-length of all other nodes. As $\rho\widehat{\ }1$ is an initial segment of η, the approximation η_s does not go to the left of $\rho\widehat{\ }1$ after stage s_0, so there is no change to m_ρ-lengths at any further point in the construction beyond s_0, and there are only the finitely many ρ-nodes in $\mathcal{A}$ that were present in $\mathcal{A}[s_0]$. Since $\mathcal{B}_{|\eta|} \cong \mathcal{A}$, there cannot be new nodes with non-minimal m_ρ-length in $\mathcal{B}_{|\eta|}[s]$ as compared to $\mathcal{B}_{|\eta|}[s_\eta]$. Nodes with the same label agree on all m-lengths at all times. So since we have an injection that is length preserving for η from $\mathcal{A}[s_\eta]$ to $\mathcal{B}_{|\eta|}[s]$, the node $f_\eta(\alpha)[s]$ must have grown into (or remained) a node with the required properties.

If α is a ρ-node with $\rho \succeq \eta\widehat{\ }0$, we will not have had any ρ-stage in between s_η and s, so this is similar to the first case.

If α is a basic node or a ρ-node with $\rho\widehat{\ }0 \preceq \eta$, then all non-zero m-lengths that should be preserved by the lemma were minimal. Any new nodes introduced into $\mathcal{A}$ between stages s_η and s would be ν-nodes with $\nu\widehat{\ }0 \preceq \eta$ or $\nu > \eta$. Such nodes have the correct m-lengths as far as α is concerned. The node $f_\eta(\alpha)$ cannot grow into an m_ν node for $\nu\widehat{\ }1 \preceq \eta$ since $\mathcal{A} \cong \mathcal{B}_{|\eta|}$ and the nodes have already been matched up. Similarly it cannot grow into an m_ν node for $\nu < \eta_s$, since the approximation will not visit left of η again and the others have been matched up. □

Lemma 6.4. *If η is on the true path and $\mathcal{A} \cong \mathcal{B}_{|\eta|}$, then $\eta\widehat{\ }0$ is on the true path.*

Proof. Let s_0 be least such that $\eta_s \geq \eta$ for all $s \geq s_0$. Suppose that $t = s_0$, or that $t > s_0$ is an $\eta\widehat{\ }0$-stage. We show that there is an $\eta\widehat{\ }0$-stage after t. Let $s > t$ be least such that s is an η-stage, $\mathcal{A}[t] \subseteq \mathcal{B}_{|\eta|}[s]$ and $\mathcal{B}_{|\eta|}[s]$ agrees with $\mathcal{A}[s-1]$ on the 0-lengths of the basic nodes. Such a stage exists since $\mathcal{A} \cong \mathcal{B}_{|\eta|}$ and since 0-lengths are finite. Consider the induced injective map h from nodes of $\mathcal{A}[t]$ to nodes of $\mathcal{B}[s]$. This map is length preserving for η, since if m is a length that must be respected then $m = m_\rho$ for some ρ with $\rho < TP(\rho \not\prec TP)$ or $\rho\widehat{\ }1 \prec \eta$. That is, there will be no more ρ-nodes after stage s_0. So if the map is not length preserving for η, then $\mathcal{A}[t]$ cannot catch up to $\mathcal{B}_{|\eta|}[s]$, contradicting $\mathcal{A} \cong \mathcal{B}_{|\eta|}$. By Lemma 6.3, we may assume h extends $f_\eta[t]$. Suppose finally that $\operatorname{dom}(f_\eta[t]) \neq \mathcal{A}[t]$ and $\operatorname{rng}(f_\eta[t]) \neq \mathcal{B}[s]$, and let β be the least node $\mathcal{B}_{|\eta|}[s] \setminus \operatorname{rng}(f_\eta[t])$. Note that $\mathcal{A}[t] \setminus \operatorname{dom}(f_\eta[t])$ consists of the nodes that were introduced into $\mathcal{A}$ at stage t. In particular, they have minimal

m-length for all m that must be respected for h to be length preserving for η. Now consider β. We may assume, increasing s if necessary, that all m-lengths that were present in $\mathcal{A}[t]$ are at least minimal in $\beta[s]$. Assume for a contradiction that h cannot be modified to still be length preserving for η and include β in its range. Then β must have a non-trivial m_ρ-length for some ρ with $\rho < TP (\rho \not\prec TP)$ or $\rho^\frown 1 \prec \eta$. But then we would have declared η inactive at stage s, contrary to Lemma 6.2. □

Lemma 6.5. *If η is on the true path and $\mathcal{A} \cong \mathcal{B}_{|\eta|}$, then f_η is onto.*

Proof. By Lemma 6.4, $\eta^\frown 0$ is on the true path. At every $\eta^\frown 0$ stage where there is a chance to extend f_η, the construction includes the next node of $\mathcal{B}$ in the range of f_η. Thus f_η is onto. □

Lemma 6.6. *The structure $\mathcal{A}$ is D-computably categorical.*

Proof. Suppose $\mathcal{A} \cong \mathcal{B}_i$. Let $\eta \prec TP$ be such that $i = |\eta|$. Let s_0 be a stage such that the approximation does not go to the left of η after stage s_0. Then the η strategy is not reset after stage s_0, so we refer to g_η^D as the version defined starting at s_0. We show that $g_\eta^D : \mathcal{A} \cong \mathcal{B}_{|\eta|}$.

For each $\alpha \in \mathcal{A}$, to compute $g_\eta^D(\alpha)$, first compute the module $U_{n,k}$ to which α belongs. If $\alpha \in U_{n,k}$, then this means the approximation to n in D changed at least k times. Let s denote the stage when the first member was introduced into $U_{n,k}$, or equivalently the stage when the approximation to n in D changed the kth time. Then $g_\eta^D(\alpha) = g_\eta^0(\alpha)$ if $D(n) = D_s(n)$, and $g_\eta^D(\alpha) = g_\eta^1(\alpha)$ otherwise.

Since the modules are disjoint, to check that g_η^D is an isomorphism, it suffices to check that it is an isomorphism on each module. Fix a module $U_{n,k}$. We have three cases to consider: No changes to the approximation of n in D after the module was started, a positive even number of changes after the module was started, or an odd number of changes since the module was started.

If there were no changes to the approximation of n in D after the module was started, then the module contains only the basic nodes μ_L and μ_R. In this case for $\alpha = \mu_L$ or $\alpha = \mu_R$, $g_\eta^D(\alpha) = g_\eta^0(\alpha) = f_\eta(\alpha)$. Since the 0-length does not get extended after the first stage it is introduced, and since α and $f_\eta(\alpha)$ have the same 0-length when $f_\eta(\alpha)$ is defined, we have $\alpha \cong f_\eta(\alpha)$ for both α, and so we have an isomorphism on the module.

If there were a positive even number of changes to the approximation of n in D after the module was started, then all 0-lengths are equal at the end of the construction. In this case $g_\eta^D(\alpha) = g_\eta^0(\alpha) = f_\eta(\alpha)$ for all α in the module. By Lemma 6.3, at recovery stages, α has the same m-lengths as $f_\eta(\alpha)$ except for m_ρ with $\rho^\frown 0 \preceq \eta$ or $\rho \geq \eta^\frown 1$, or 0. Now for ρ with $\rho^\frown 0 \preceq \eta$ we have that the m_ρ lengths are all infinite at the end of the construction. For ρ with $\rho^\frown 0 \preceq \eta$, the ρ strategies always get canceled with discarded m_ρ-lengths equalized. All 0-lengths are equal. Thus each $\alpha \cong f_\eta(\alpha)$. The map f_η is easily seen to be injective by its construction, and is surjective by Lemma 6.5.

Finally, consider the case when there were an odd number of changes to the approximation of n in D after the module was started. In this case $g_\eta^D(\alpha) = g_\eta^1(\alpha)$ for all α in the module. For ρ-nodes α with $\rho^\frown 0 \not\preceq \eta^\frown 0$ we have $g_\eta^1(\alpha) = f_\eta(\alpha)$. For these nodes, Lemma 6.3 also guarantees equality of 0-lengths between α and $f_\eta(\alpha)$ at recovery stages. So by an argument as in the previous paragraph,

$\alpha \cong f_\eta(\alpha)$ for these α. For α a basic node or a ρ-node with $\rho^\frown 0 \preceq \eta$ and $l \geq 1$, we have $g^1_\eta(\mathbb{F}^{l-1}_\eta(\alpha)) = f_\eta(\mathbb{F}^l_\eta(\alpha))$, and $g^1_\eta(\mathbb{B}^l_\eta(\alpha)) = f_\eta(\mathbb{B}^{l-1}_\eta(\alpha))$. Recall that all η nodes belong to the forward and backward sequences for such α, so these definitions complete the definition of g^1_η on all nodes. Now $\mathbb{F}^l_\eta(\alpha)$ is an η-node, so by Lemma 6.3 and our usual argument, $\mathbb{F}^l_\eta(\alpha) \cong f_\eta(\mathbb{F}^l_\eta(\alpha))$. According to step 3 of the construction, each node in the forward sequence for α has the same 0-length as α, and indeed at any stage the members of the forward sequence for α differ from α and each other only in their m_η-length. Since $\eta^\frown 0$ is on the true path, the m_η-lengths are all infinite. Thus $\mathbb{F}^{l-1}_\eta(\alpha)) \cong \mathbb{F}^l_\eta(\alpha) \cong f_\eta(\mathbb{F}^l_\eta(\alpha)) = g^1_\eta(\mathbb{F}^{l-1}_\eta(\alpha))$. For nodes of the backward sequence for α, the 0-length of each is equal to that of $f_\eta(\alpha)$, and they differ from each other only on their m_η-length. For $l \geq 2$, Lemma 6.3 and our usual argument gives $\mathbb{B}^{l-1}_\eta(\alpha) \cong f_\eta(\mathbb{B}^{l-1}_\eta(\alpha))$, so that at the end of the construction $\mathbb{B}^l_\eta(\alpha)) \cong \mathbb{B}^{l-1}_\eta(\alpha) \cong f_\eta(\mathbb{B}^{l-1}_\eta(\alpha)) = g^1_\eta(\mathbb{B}^l_\eta(\alpha))$. Also by Lemma 6.3, α agrees with $f_\eta(\alpha)$ on all important lengths except possibly on the 0-length. Since the 0-length of $\mathbb{B}^l_\eta(\alpha))$ is equal to that of $f_\eta(\alpha)$, it follows that at the end of the construction $\mathbb{B}^1_\eta(\alpha)) \cong f_\eta(\alpha) = g^1_\eta(\mathbb{B}^l_\eta(\alpha))$. The map g^1_η is a bijection since f_η is. Indeed, it exactly copies f_η on nodes with $\rho^\frown 0 \not\preceq \eta^\frown 0$. The rest of the nodes all belong to disjoint forward/backward sequences, on which the f_η map is shifted by one along the sequence by g^1_η. □

Lemma 6.7. *$\mathcal{A}$ and $\mathcal{A}^*$ are computable copies of the same structure.*

Proof. They are the same stage-by-stage. There is only one stage in each disjoint module where the action taken is not exactly the same, so they are isomorphic in the limit. □

Lemma 6.8. *If $f : \mathcal{A} \cong \mathcal{A}^*$, then $f \geq_T D$.*

Proof. The procedure for computing whether $n \in D$ is as follows. Find the least k such that $f(\mu_L) = \mu_L$ for μ_L in the module $U_{n,k}$. Then $n \in D \iff n \in D_k$. □

7. Questions

Many open questions about degrees of categoricity remain; here are our two favourites.

Bazhenov, Kalimullin and Yamaleev [BKY16], and Csima and Stephenson [CS19] gave examples of structures whose degree of categoricity is not a strong degree for that structure. However, we do not know if every degree of categoricity is a strong degree of categoricity for some structure:

Question 7.1. *Are all degrees of categoricity strong?*

The result of this paper shows that one will need to look outside the Δ^0_2 degrees for a counterexample.

Question 7.2. *If $\mathbf{d}$ is such that $\mathbf{0}^{(\alpha)} \leq \mathbf{d} \leq \mathbf{0}^{(\alpha+1)}$ for some computable ordinal α, must $\mathbf{d}$ be a degree of categoricity?*

It seems likely that the techniques of this paper, combined with the techniques in [CDHTM20] could be used to give a positive answer to this question. Thus, we conjecture that this holds in the positive.

References

[AC16] Bernard Anderson and Barbara Csima. Degrees that are not degrees of categoricity. *Notre Dame J. Form. Log.*, 57(3):389–398, 2016.

[BKY16] N. A. Bazhenov, I. Sh. Kalimullin, and M. M. Yamaleev. On strong and not-strong degrees of categoricity. *Algebra Logika*, 55(2):257–263, 2016.

[BKY18] N. A. Bazhenov, I. Sh. Kalimullin, and M. M. Yamaleev. Degrees of categoricity and spectral dimension. *J. Symb. Log.*, 83(1):103–116, 2018.

[BM] N. Bazhenov and M. Marchuk. Degrees of categoricity for prime and homogeneous models. Preprint.

[CDHTM20] Barbara F. Csima, Michael Deveau, Matthew Harrison-Trainor, and Mohammad Assem Mahmoud. Degrees of categoricity above limit ordinals. *Computability*, 9(2):127–137, 2020.

[CFS13] B. F. Csima, J. N. Y. Franklin, and R. A. Shore. Degrees of categoricity and the hyperarithmetic hierarchy. *Notre Dame J. Form. Log.*, 54(2):215–231, 2013.

[CS19] B. F. Csima and J. Stephenson. Finite computable dimension and degrees of categoricity. *Annals of Pure and Applied Logic*, 170(1):58–94, 2019.

[FKM10] E. B. Fokina, I. Kalimullin, and R. Miller. Degrees of categoricity of computable structures. *Arch. Math. Logic*, 49(1):51–67, 2010.

[Fra17] Johanna N. Y. Franklin. Strength and weakness in computable structure theory. In *Computability and complexity*, volume 10010 of *Lecture Notes in Comput. Sci.*, pages 302–323. Springer, Cham, 2017.

[Gon11] S. S. Goncharov. Degrees of autostability relative to strong constructivizations. *Tr. Mat. Inst. Steklova*, 274(Algoritmicheskie Voprosy Algebry i Logiki):119–129, 2011.