EXTENSIONS OF CATEGORICITY RELATIVE TO A DEGREE

JAVA DARLEEN VILLANO

ABSTRACT. In this paper, we apply the machinery developed in [Vil25] to study the behavior of computable categoricity relativized to non-c.e. degrees. In particular, we show that we can build a computable structure which is not computably categorical but is computably categorical relative to a 1-generic degree. Additionally, we show that other classes of structures besides directed graphs admit a computable example which can change its computable categorical behavior relative to different degrees.

1. INTRODUCTION

In computable structure theory, we are interested in effectivizing model theoretic notions and constructions. For a general background on computable structure theory, see Ash and Knight [Ash00] or Montalbán [Mon21]. In particular, many people have examined the complexity of isomorphisms between structures within the same isomorphism type. We restrict ourselves to countable structures in a computable language and assume their domain is ω.

A computable structure $\mathcal{A}$ is *computably categorical* if for any computable copy $\mathcal{B}$ of $\mathcal{A}$, there exists a computable isomorphism between $\mathcal{A}$ and $\mathcal{B}$. In this paper, we continue the investigation started in [DHTM21] and [Vil25] into the following relativized version of computable categoricity. For $X \in 2^{\mathbb{N}}$, we say that a computable structure $\mathcal{A}$ is *computably categorical relative to X* if for all X-computable copies $\mathcal{B}$ of $\mathcal{A}$, there exists an X-computable isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

By relativized versions of results due to Ash, Knight, Manasse, and Slaman [Ash+89], Chisholm [Chi90], and Goncharov [Gon80], we have for every computable structure $\mathcal{A}$ that this relativization stabilizes on a cone above $\mathbf{0}''$. That is, either $\mathcal{A}$ is computably categorical relative to all degrees above $\mathbf{0}''$, or to no degree above $\mathbf{0}''$. Downey, Harrison-Trainor, Melnikov in [DHTM21] and the author in [Vil25] later showed that in the c.e. degrees, this relativization behaves chaotically in the following sense.

Theorem 1.1 (Downey, Harrison-Trainor, Melnikov [DHTM21]). *There is a computable structure $\mathcal{A}$ and c.e. degrees $0 = Y_0 <_T X_0 <_T Y_1 <_T X_1 <_T \ldots$ such that*

(1) $\mathcal{A}$ *is computably categorical relative to Y_i for each i,*
(2) $\mathcal{A}$ *is not computably categorical relative to X_i for each i,*
(3) $\mathcal{A}$ *is relatively computably categorical to $\mathbf{0}'$.*

Theorem 1.2 ([Vil25]). *Let $P = (P, \leq)$ be a computable partially ordered set and let $P = P_0 \sqcup P_1$ be a computable partition. Then, there exists a computable computably categorical directed graph $\mathcal{G}$ and an embedding h of P into the c.e. degrees where $\mathcal{G}$ is computably categorical relative to each degree in $h(P_0)$ and is not computably categorical relative to each degree in $h(P_1)$.*

We now discuss some restrictions of the techniques in [Vil25]. We first introduce the following definitions.

Definition 1.3. A computable structure $\mathcal{A}$ is **d**-*computably categorical* if for all computable $\mathcal{B} \cong \mathcal{A}$, there exists a **d**-computable isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Definition 1.4. A structure $\mathcal{A}$ has *degree of categoricity* **d** if $\mathcal{A}$ is **d**-computably categorical and for all **c**, if $\mathcal{A}$ is **c**-computably categorical, then $\mathbf{d} \leq \mathbf{c}$. A degree **d** is a *degree of categoricity* if there is some structure with degree of categoricity **d**.

Finding a characterization of degrees of categoricity in the Turing degrees has recently been an active topic in computable structure theory. For a survey paper of development up until 2017, see Franklin [Fra17]. Degrees which are *not* degrees of categoricity exist, with Anderson and Csima producing several examples in [AC16]. One important example is the following.

Theorem 1.5 (Anderson, Csima [AC16]). *There is a Σ_2^0 degree that is not a degree of categoricity.*

In fact, the Σ_2^0 degree that they built to witness this result is low for isomorphism.

Definition 1.6. A degree **d** is *low for isomorphism* if for every pair of computable structures $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A}$ is **d**-computably isomorphic to $\mathcal{B}$ if and only if $\mathcal{A}$ is computably isomorphic to $\mathcal{B}$.

There are currently no known characterizations for LFI degrees, but they are connected to the generic degrees. We quickly recall the definition of an n-generic set.

Definition 1.7. A set A is n-*generic* if for all Σ_n^0 set of strings $S \subseteq 2^{<\omega}$, there exists an m such that either $A \restriction m \in S$ or for all $\tau \supseteq A \restriction m$, $\tau \notin S$. A degree **d** is n-*generic* if it contains an n-generic set.

Theorem 1.8 (Franklin, Solomon [FS14]). *Every 2-generic degree is low for isomorphism.*

This tells us that for a 2-generic degree **d**, there exists *no* computable structure $\mathcal{A}$ where $\mathcal{A}$ is not computably categorical but is computably categorical relative to **d**, one of the first examples of a degree where techniques from [Vil25] do not apply. Here, we will show that this is optimal in the generic degrees, that is, we can build a 1-generic degree **d** and a computable structure $\mathcal{A}$ which is not computably categorical but is computably categorical relative to **d**.

Theorem 1.9. *There exists a (properly) 1-generic G such that there is a computable directed graph $\mathcal{A}$ where $\mathcal{A}$ is not computably categorical but is computably categorical relative to G.*

The proof is a priority construction on a tree of strategies, using largely the same machinery developed in [Vil25]. In Section 2, we introduce informal descriptions for the strategies we need to satisfy our requirements for the construction and discuss important interactions between certain strategies. In Section 3, we detail the formal strategies, state and prove

auxiliary lemmas about our construction, and state and prove the main verification lemma. We then end this paper with Section 4 where we discuss for which classes of structures admit or do not admit a computable example that can change its categorical behavior relative to a degree above $\mathbf{0}$.

2. Informal strategies for Theorem 1.9

We first establish the informal strategies to meet three goals for our construction: to build a 1-generic set G, to make our graph $\mathcal{A}$ not computably categorical, and to make our graph $\mathcal{A}$ computably categorical relative to G. We will then describe the interactions that occur when we use all three strategies together, and how to resolve any issues that arise.

2.1. Building a 1-generic G.

We will build a 1-generic G via infinitely many strategies. Recall that for a set G to be 1-generic, we must have that for all $e \in \omega$, there exists an initial segment σ of G where either $\sigma \in W_e$ or for all extensions $\tau \supseteq \sigma$, $\tau \notin W_e$. That is, either G meets or avoids each c.e. set.

For each $j \in \omega$, we meet the requirement

$$R_j : (\exists \sigma \subseteq G)(\sigma \in W_j \vee (\forall \tau \supseteq \sigma)(\tau \notin W_j)).$$

We will define sets $G[s]$, where s is a stage number, which are a Δ_2^0 approximation to our 1-generic G. Let α be the highest priority R-strategy. When it is first eligible to act at a stage s_0, α will define a parameter $n_\alpha > 0$ large (and so in particular, $n_\alpha > s_0$), and its goal is to find an extension $\tau_0 \supseteq G[s_0] \upharpoonright n_\alpha$ where $\tau_0 \in W_j$. At each α-stage, it searches for such an extension.

If α never finds such an extension, then we have that all extensions of $G[s_0] \upharpoonright n_\alpha$ avoid W_j and α succeeds trivially. If α finds an extension τ_0 such that $\tau_0 \in W_j$ at a stage $s_1 \geq s_0$, then α will define $G[s_1] = \tau_0 {}^\frown 0^\omega$ and will initialize all lower priority strategies. In particular, all lower priority R-strategies will now have to redefine new, larger parameters (and so these parameters will be bigger than $|\tau_0|$) and must now use $G[s_1]$ as the current approximation of G at the end of stage s_1 for all stages $s \geq s_1$. We also have that $G[s_1] \upharpoonright n_\alpha = G[s_0] \upharpoonright n_\alpha$, so α does not cause changes on numbers below n_α.

If all R_j-strategies succeeded, then we define $G = \lim_{s \to \infty} G[s]$ to be our 1-generic set. This limit exists by the observation in the previous paragraph.

2.2. Making $\mathcal{A}$ not computably categorical.

We will build our directed graph $\mathcal{A}$ in stages. At stage $s = 0$, we set $\mathcal{A} = \emptyset$. Then, at stage $s > 0$, we add two new connected components to $\mathcal{A}[s]$ by adding the root nodes a_{2s} and a_{2s+1} for those components, and attaching to each node a 2-loop (a cycle of length 2). We then attach a $(5s + 1)$-loop to a_{2s} and a $(5s + 2)$-loop to a_{2s+1}. This gives us the configuration of loops:

$$a_{2s} : 2, 5s + 1$$
$$a_{2s+1} : 2, 5s + 2.$$

The connected component consisting of the root node a_{2s} with its attached loops will be referred to as the *2sth connected component* of $\mathcal{A}$. During the construction, we might add more loops to connected components of $\mathcal{A}$, which causes them to have the following configuration:

$$a_{2s} : 2, 5s + 1, 5s + 2, 5s + 3$$
$$a_{2s+1} : 2, 5s + 1, 5s + 2, 5s + 4$$

This configuration, if certain interactions occur during the construction, can also transform into the following:

$$a_{2s} : 2, 5s+1, 5s+2, 5s+3, 5s+4$$
$$a_{2s+1} : 2, 5s+1, 5s+2, 5s+3, 5s+4.$$

Note that the last configuration has that the $2s$th and $(2s+1)$st components of $\mathcal{A}$ are isomorphic, which may be necessary as a result of a special interaction between strategies of all three types of requirements in this construction (see 2.4).

In order to make $\mathcal{A}$ not computably categorical, it is sufficient to construct a computable copy $\mathcal{B}$ such that for all $e \in \omega$, the computable function Φ_e is not an isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Similarly to $\mathcal{A}$, we build the directed graph $\mathcal{B}$ in stages. At stage $s = 0$, we set $\mathcal{B} = \emptyset$. At stage $s > 0$, we add root nodes b_{2s} and b_{2s+1} to $\mathcal{B}$ and attach to each one a 2-loop. Next, we attach a $(5s+1)$-loop to b_{2s} and a $(5s+2)$-loop to b_{2s+1}. However, throughout the construction, we may add new loops to specific components of $\mathcal{B}$. For the $2s$th and $(2s+1)$st components of $\mathcal{B}$, we have three possible final configurations of the loops. If we never start the process of diagonalizing using these components, then they will remain the same forever:

$$b_{2s} : 2, 5s+1$$
$$b_{2s+1} : 2, 5s+2.$$

If we use these components to diagonalize against a computable map Φ_e, they will end in the following configuration:

$$b_{2s} : 2, 5s+1, 5s+2, 5s+4$$
$$b_{2s+1} : 2, 5s+1, 5s+2, 5s+3.$$

If these components were used to diagonalize against Φ_e and then certain interactions occur between all three types of requirements, these components may have the final configuration:

$$b_{2s} : 2, 5s+1, 5s+2, 5s+3, 5s+4$$
$$b_{2s+1} : 2, 5s+1, 5s+2, 5s+3, 5s+4.$$

For all $e \in \omega$, we meet the following requirement

$$P_e : \Phi_e : \mathcal{A} \to \mathcal{B} \text{ is not an isomorphism.}$$

To satisfy this requirement, we will diagonalize against Φ_e. Let α be a P_e-strategy.

When α is first eligible to act, it picks a large number n_α, and for the rest of this strategy, let $n = n_\alpha$. This parameter indicates which connected components of $\mathcal{B}$ will be used in the diagonalization. At future stages, α checks if Φ_e maps the $2n$th and $(2n+1)$st connected component of $\mathcal{A}$ to the $2n$th and $(2n+1)$st connected component of $\mathcal{B}$, respectively. If not, α does not take any action. If α sees such a computation, it acts in the following way.

At this point, our connected components in $\mathcal{A}[s]$ and $\mathcal{B}[s]$ are as follows:

$$a_{2n} : 2, 5n+1 \qquad b_{2n} : 2, 5n+1$$
$$a_{2n+1} : 2, 5n+2 \qquad b_{2n+1} : 2, 5n+2.$$

Since Φ_e looks like a potential isomorphism between $\mathcal{A}$ and $\mathcal{B}$, α will now take action to eventually force the true isomorphism to match a_{2n} with b_{2n+1} and to match a_{2n+1} with b_{2n}.

α adds a $(5n+2)$- and $(5n+3)$-loop to a_{2n} and a $(5n+1)$- and $(5n+4)$-loop to a_{2n+1} in $\mathcal{A}[s]$. It also attaches a $(5n+2)$- and $(5n+4)$-loop to b_{2n} and a $(5n+1)$- and $(5n+3)$-loop to b_{2n+1} in $\mathcal{B}[s]$. Our connected components in $\mathcal{A}[s]$ and in $\mathcal{B}[s]$ are now:

$$a_{2n} : 2, 5n+1, 5n+2, 5n+3 \quad b_{2n} : 2, 5n+1, 5n+2, 5n+4$$
$$a_{2n+1} : 2, 5n+1, 5n+2, 5n+4 \quad b_{2n+1} : 2, 5n+1, 5n+2, 5n+3.$$

For all higher priority S-strategies β such that $\beta^\frown\langle\infty\rangle \subseteq \alpha$, α enumerates the use u_{β,n_α} into G and sets $n_\beta = n_\alpha$. This enumeration occurs because if we have that $\beta^\frown\langle\infty\rangle \subseteq \alpha$, then α believes β will define a total function f_β^G on $\mathcal{A}$. Therefore, α doesn't restart when β extends its definition of f_β^G and so this map may be defined on the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}$ when α acts. In this case, $f_\beta^G[s-1]$ maps the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}$ to their copies in $\mathcal{M}_i^G$ with some use u_{β,n_α}. We will choose this use carefully so that we know $u_{\beta,n_\alpha} \notin G[s-1]$, and in fact, $u_{\beta,n_\alpha} \notin G[t]$ for all $t < s$. This allows α to enumerate the use u_{β,n_α} into $G[s]$ to destroy the f_β^G computation on the $2n_\alpha$th and $(2n_\alpha+1)$st components in $\mathcal{A}$.

We will refer to these two actions as α *issuing a challenge* to β. α now takes the outcome w_2. If by the next α-stage we have that α has not been initialized, then it takes the success outcome s. By α's actions above, we have that $\Phi_e(a_{2n}) = b_{2n}$ and $\Phi_e(a_{2n+1}) = b_{2n+1}$, and so Φ_e fails to be a computable isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

2.3. **Being computably categorical relative to G.** Since we want to make $\mathcal{A}$ computably categorical relative to our 1-generic G, we must define embeddings using G as an oracle. Additionally, we are looking at (partial) G-computable directed graphs $\mathcal{M}_i^G$ with domain ω whose edge relation is given by Φ_i^G. Any changes in initial segments of G can cause changes for both our partial G-computable embeddings and in $\mathcal{M}_i^G$ throughout the construction.

We define the following terms to keep track of certain finite subgraphs which appear and may remain in $\mathcal{M}_i^G$ throughout our construction.

Definition 2.1. Let C_0 and C_1 be isomorphic finite distinct subgraphs of $\mathcal{M}_i^G[s]$. The *age of* C_0 is the least stage $t \leq s$ such that all edges in C_0 appear in $\mathcal{M}_i^G[t]$ and remained in $\mathcal{M}_i^G[s']$ for all $t \leq s' \leq s$, denoted by $\mathrm{age}(C_0)$. We say that C_0 is *older than* C_1 when $\mathrm{age}(C_0) \leq \mathrm{age}(C_1)$.

We say that C_0 is the *oldest* if for all finite distinct subgraphs $C \cong C_0$ of $\mathcal{M}_i^G[s]$, $\mathrm{age}(C_0) \leq \mathrm{age}(C)$.

Definition 2.2. Let $C_0 = \langle a_0, a_1, \ldots, a_k \rangle$ and $C_1 = \langle b_0, b_1, \ldots, b_k \rangle$ be isomorphic finite distinct subgraphs of $\mathcal{M}_i^G[s]$ with $a_0 < a_1 < \cdots < a_k$ and $b_0 < b_1 < \cdots < b_k$. We say that $C_0 <_{\mathrm{lex}} C_1$ if for the least j such that $a_j \neq b_j$, $a_j < b_j$.

We say that C_0 is the *lexicographically least* if for all finite distinct subgraphs $C \cong C_0$ of $\mathcal{M}_i^G[s]$, $C_0 <_{\mathrm{lex}} C$.

If $\mathcal{A} \cong \mathcal{M}_i^G$, then we need to build a G-computable isomorphism between these graphs. For each $i \in \omega$, we meet the following requirement

S_i : if $\mathcal{A} \cong \mathcal{M}_i^G$, then there exists a G-computable isomorphism $f_i^G : \mathcal{A} \to \mathcal{M}_i^G$.

We will show in the verification that if $\mathcal{A} \cong \mathcal{M}_i^G$, "true" copies of components from $\mathcal{A}$ will eventually appear and remain in $\mathcal{M}_i^G$ (and thus become the oldest and lex-least finite subgraph which is isomorphic to a component in $\mathcal{A}$), and so our S_i-strategy below will be able to define the correct G-computable isomorphism between the two graphs.

Let α be an S_i-strategy. When α is first eligible to act, it sets its parameter $n_\alpha = 0$ and defines f_α^G to be the empty map. Once α has defined n_α, then when α acted at the previous α-stage s_0, we have the following situation:

(1) For each $m < n_\alpha$, $f_\alpha^G[s_0]$ maps the $2m$th and $(2m+1)$st components of $\mathcal{A}[s_0]$ to isomorphic copies in $\mathcal{M}_i^G[s_0]$.

(2) For $m < n_\alpha$, let l_m be the maximum $\Phi_i^G[s_0]$-use for the loops in the copies in $\mathcal{M}_i^G[s_0]$ for the $2m$th and $(2m+1)$st components in $\mathcal{A}$. We can assume that if $m_0 < m_1 < n_\alpha$, then $l_{m_0} < l_{m_1}$.

(3) For $m < n_\alpha$, let $u_{\alpha,m}$ be the $f_\alpha^G[s_0]$-use for the mapping of the $2m$th and $(2m+1)$st components of $\mathcal{A}$. This use will be constant for all elements in these components.

(4) By construction, we will have that $l_m < u_{\alpha,k}$ for all $m \leq k < n_\alpha$.

Suppose α is acting at stage s and that $n_\alpha > 0$ and let s_0 be the previous α-stage in the construction. We now break into cases.

If α took an outcome w_n at s_0, then no strategy could have changed G below the use $u_{\alpha,n_\alpha-1}$, and so the value of n_α remains unchanged.

If instead we have that α took the ∞ outcome at s_0, then G may have changed underneath $u_{\alpha,n_\alpha-1}$. We will show in the verification that the only strategies that can change G below this use are P or R-strategies β such that $\beta \supseteq \alpha^\frown\langle\infty\rangle$. If $G[s] \upharpoonright (u_{\alpha,n_\alpha-1} + 1) \neq G[s_0] \upharpoonright (u_{\alpha,n_\alpha-1}+1)$, then some $\beta \supseteq \alpha^\frown\langle\infty\rangle$ caused a change at stage s_0 after α acted. Furthermore, at most one such β can cause this change at stage s_0, as every other strategy extending $\beta^\frown\langle s\rangle$ will define new and large witnesses for the remainder of stage s_0.

If β is a P-strategy which changed G under $u_{\alpha,n_\alpha-1}$, then it added diagonalizing loops to the $2n_\beta$th and $(2n_\beta + 1)$st components in $\mathcal{A}$. Assuming that $n_\beta < n_\alpha$, then α has already defined $f_\alpha^G[s_0]$ on these components. Hence, β enumerates the use u_{α,n_β} into G to destroy this computation. To account for this change, α redefines n_α to be the least m such that f_α^G is not currently defined on the $2m$th and $(2m+1)$st components of $\mathcal{G}$.

If instead we have that the β is an R-strategy, then suppose β redefines G by setting $G[s_0] = \tau^\frown 0^\omega$. This definition may have changed G on numbers as small as n_α and hence may have injured previously defined f_α^G computations. So, at stage s, α resets n_α to be the least m such that f_α^G is not currently defined on the $2m$th and $(2m+1)$st components of $\mathcal{G}$.

We now carry out the main action of the S_i-strategy: we check whether we can extend $f_\alpha^G[s-1]$ to the $2n_\alpha$th and $(2n_\alpha + 1)$st components of $\mathcal{A}[s]$. Search for isomorphic copies in $\mathcal{M}_i^G[s]$ of these components. If there are multiple copies in $\mathcal{M}_i^G[s]$, choose the oldest such copy to map to, and if there are multiple equally old copies, choose the lexicographically least oldest copy. If there are no copies in $\mathcal{M}_i^G[s]$, then keep the value of n_α the same, f_α^G unchanged, and let the next requirement act. Otherwise, extend $f_\alpha^G[s-1]$ to $f_\alpha^G[s]$ to include the $2n_\alpha$th and $(2n_\alpha + 1)$st components of $\mathcal{A}$ with a large use u_{α,n_α}. Since u_{α,n_α} was chosen large, we have that $u_{\alpha,n_\alpha} > l_k$ for all $k \leq n_\alpha$. Increment n_α by 1 and go to the next requirement. If α had to redefine n_α because it was challenged by an S-strategy β of higher priority, then α continues to take finitary outcomes until the value of n_α exceeds n_β.

If $\mathcal{A} \cong \mathcal{M}_i^G$, then for each n, eventually the real copies of the $2n$th and $(2n+1)$st components of $\mathcal{A}$ will appear in $\mathcal{M}_i^G$. Moreover, they will eventually be the oldest and lexicographically least copies in $\mathcal{M}_i^G$. After G has settled down on the the maximum true G-use on the edges in the loops in these $\mathcal{M}_i^G$ components, we will define f_α^G correctly on these components with a large use $u_{\alpha,n}$. Therefore, eventually our map f_α^G is never injured again on the $2n$th and $(2n+1)$st components. It follows that if $A \cong \mathcal{M}_i^G$, then f_α^G will be an

embedding of $\mathcal{A}$ into $\mathcal{M}_i^G$ which can be extended to a G-computable isomorphism defined on all of $\mathcal{A}$.

2.4. An interaction caused by genericity.

The following interaction arises because we are building a 1-generic. The fact that initial segments of G can change several times throughout the construction requires our isolated strategies to be more flexible than what was done previously in [Vil25], and we outline the needed changes to affected strategies below.

Let α be an S_i-strategy, β be an R_j-strategy, and γ be a P_e-strategy where $\alpha^\frown\langle\infty\rangle \subseteq \beta^\frown\langle w_1\rangle \subseteq \gamma$, where $\beta^\frown\langle w_1\rangle$ indicates that β defined its parameter n_β and took the waiting outcome at the last β-stage. Suppose that n_β, at a stage s_0, and n_γ have been defined and that $n_\beta < n_\gamma$. Suppose that at stage s_1, α is able to define $f_\alpha^G[s_1]$ with a use u_{α,n_γ} on the $2n_\gamma$th and $(2n_\gamma + 1)$st components of $\mathcal{A}$, with the configuration of the loops in $\mathcal{A}[s_1]$ and $\mathcal{M}_i^G[s_1]$ being:

$$a_{2n_\gamma} : 2, 5n_\gamma + 1 \quad c : 2, 5n_\gamma + 1$$
$$a_{2n_\gamma+1} : 2, 5n_\gamma + 2 \quad d : 2, 5n_\gamma + 2.$$

Note that $f_\alpha^G[s_1]$ on these two $\mathcal{A}$-components is being protected by the initial segment $G[s_1] \restriction (u_{\alpha,n_\gamma} + 1)$. Then, at the end of stage s_1, α takes the ∞ outcome, and β is eligible to act. Suppose that β continues to take the w_1 outcome, and so γ is eligible to act at a stage $s_2 > s_1$. Suppose that γ sees that the map $\Phi_e[s_2]$ maps its chosen $\mathcal{A}$-components isomorphically to their copies in $\mathcal{B}[s_2]$, and thus begins to diagonalize by adding new loops to the following $\mathcal{A}$-components and $\mathcal{B}$-components:

$$a_{2n_\gamma} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3 \quad b_{2n_\gamma} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 4$$
$$a_{2n_\gamma+1} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 4 \quad b_{2n_\gamma+1} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3.$$

When γ adds these loops, it enumerates u_{α,n_γ} into $G[s_2]$, and so $G[s_2] \restriction (u_{\alpha,n_\gamma} + 1) \neq G[s_1] \restriction (u_{\alpha,n_\gamma} + 1)$ and $f_\alpha^G[s_1]$ disappears on the the $2n_\gamma$th and $(2n_\gamma + 1)$st components of $\mathcal{A}$. Let $s_3 \geq s_2$ be a stage such that newly added loops first appeared in $\mathcal{M}_i^G[s_3]$ in the following positions:

$$a_{2n_\gamma} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3 \quad c : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 4$$
$$a_{2n_\gamma+1} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 4 \quad d : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3.$$

α is now able to recover its map f_α^G on these $\mathcal{A}$-components with a new large use u'_{α,n_γ} by mapping a_{2n_γ} to d, $a_{2n_\gamma+1}$ to c, and all nodes in each component to their respective copies in $\mathcal{M}_i^G[s_3]$. Note that this new map $f_\alpha^G[s_3]$ on these $\mathcal{A}$-components is being protected by $G[s_3] \restriction (u'_{\alpha,n_\gamma} + 1)$. At the end of stage s_3, α takes the ∞ outcome and suppose β is now eligible to act again. Now, suppose that β has found an extension τ_{n_β} of $G[s_0] \restriction n_\beta$ which is in $W_j[s_4]$ where $s_4 > s_3$. β then extends $G[s_0] \restriction n_\beta$ to τ_{n_β} and defines $G[s_4] = \tau_{n_\beta}^\frown 0^\omega$. Furthermore, suppose that $G[s_4] \restriction (u_{\alpha,n_\gamma} + 1) = G[s_1] \restriction (u_{\alpha,n_\gamma} + 1)$, and so the original map $f_\alpha^G[s_1]$ is restored on the $2n_\gamma$th and $(2n_\gamma + 1)$st components of $\mathcal{A}$. β taking the success outcome initializes γ since $\gamma \supseteq \beta^\frown\langle w_1\rangle$, however because $\mathcal{A}$ and $\mathcal{B}$ need to be computable directed graphs, the loops added at stage s_2 must remain in both graphs. In particular, if

the true outcome for $\mathcal{M}_i^G$ is to have the following connected components

$$c : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 4$$
$$d : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3,$$

then because $f_\alpha^G[s_1]$ was restored on the two $\mathcal{A}$-components, it is now incorrect on the G-computable embedding from $\mathcal{A}$ into $\mathcal{M}_i^G$ since $f_\alpha^G[s_1](a_{2n_\gamma}) = c$ and $f_\alpha^G[s_1](a_{2n_\gamma+1}) = d$, in particular:

$$a_{2n_\gamma} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3 \quad c : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 4$$
$$a_{2n_\gamma+1} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 4 \quad d : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3.$$

We have then that if $\alpha^\frown \langle \infty \rangle$ is on the true path, it will define an incorrect G-computable embedding from $\mathcal{A}$ into $\mathcal{M}_i^G$. To resolve this issue, after β finds its extension, we will add an additional step to the R_j-strategy where we homogenize any affected $\mathcal{A}$-components defined so far, namely the $2n_\gamma$th and $(2n_\gamma + 1)st$ components of $\mathcal{A}$ in this example, in the following way:

$$a_{2n_\gamma} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3, 5n_\gamma + 4$$
$$a_{2n_\gamma+1} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3, 5n_\gamma + 4.$$

Then, if the newly added loops reappear in the corresponding copies of each component in $\mathcal{M}_i^G$ and remain, it does not matter which positions they appear in since $f_\alpha^G[s_1]$ will be correct no matter what. If other cycles or the loops never appear again after β's success, then we have that $\mathcal{A} \not\cong \mathcal{M}_i^G$ and so α trivially wins. Furthermore, the path now goes to the left of γ, initializing γ, and so we do not hurt our construction by undoing γ's diagonalization by homogenizing.

Additionally, since we need that $\mathcal{A} \cong \mathcal{B}$, we must also homogenize the corresponding components in $\mathcal{B}$ as well:

$$b_{2n_\gamma} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3, 5n_\gamma + 4$$
$$b_{2n_\gamma+1} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3, 5n_\gamma + 4.$$

Lastly, we will also homogenize the $\mathcal{A}$-components which received diagonalizing loops from P-strategies which are to the right of the current true path.

3. Proof of Theorem 1.9

3.1. Requirements. We have three requirements for our construction:

$$R_j : (\exists \sigma \subseteq G)(\sigma \in W_j \vee (\forall \tau \supseteq \sigma)(\tau \notin W_j));$$
$$P_e : \Phi_e : \mathcal{A} \to \mathcal{B} \text{ is not an isomorphism;}$$
$$S_i : \text{if } \mathcal{A} \cong \mathcal{M}_i^G, \text{ then there exists a } G\text{-computable isomorphism } f_i^G : \mathcal{A} \to \mathcal{M}_i^G.$$

3.2. Construction. Let $\Lambda = \{\infty <_\Lambda \cdots <_\Lambda s <_\Lambda w_2 <_\Lambda w_1 <_\Lambda w_0\}$ be the set of outcomes, and let $T = \Lambda^{<\omega}$ be our tree of strategies. The construction will be performed in ω many stages s.

We define the *current true path* π_s, the longest strategy eligible to act at stage s, inductively. For every s, λ, the empty string, is eligible to act at stage s. Suppose the strategy α is eligible to act at stage s. If $|\alpha| < s$, then follow the action of α to choose a successor $\alpha^\frown \langle o \rangle$ on the current true path. If $|\alpha| = s$, then set $\pi_s = \alpha$. For all strategies β such that $\pi_s <_L \beta$, initialize β (that is, set all parameters associated to β to be undefined). Also, if

$\pi_s <_L \beta$ and β is a P-strategy such that the $2n_\beta$th and $(2n_\beta + 1)$st components of $\mathcal{A}$ and $\mathcal{B}$ have been diagonalized (that is, the $(5n + 3)$- and $(5n + 4)$-loops have been added), then homogenize these components in $\mathcal{A}$ and $\mathcal{B}$. If $\beta <_L \pi_s$ and $|\beta| < s$, then β retains the same values for its parameters.

We will now give formal descriptions of each strategy and their outcomes in the construction.

3.3. R_j-strategies and outcomes.

We first cover the R_j-strategies used to build our 1-generic set G. Let α be an R_j-strategy eligible to act at stage s.

Case 1: If α is acting for the first time at stage s or has been initialized since the last α-stage, it defines $m_\alpha = \max\{n_\beta : n_\beta$ defined for $\beta \subset \alpha\}$ only when n_β has been defined for all $\beta \subset \alpha$. Once m_α has been defined, α waits to see if f_γ^G converges on the $2m_\alpha$th and $(2m_\alpha + 1)$st components of $\mathcal{A}$ for all S-strategies $\gamma^\frown \langle \infty \rangle \subseteq \alpha$. Until f_γ^G converges on all of those components, it remains in *Case 1* by taking the w_0 outcome and does not define its parameter n_α.

Case 2: If α took the w_0 outcome at the previous α-stage and all maps f_γ^G have converged on the $2m_\alpha$th and $(2m_\alpha + 1)$st components of $\mathcal{A}$ for S-strategies $\gamma^\frown \langle \infty \rangle \subseteq \alpha$, it defines its parameter n_α to be large. In particular, n_α is greater than all of the uses of f_γ^G on the $2n_\beta$th and $(2n_\beta + 1)$st components for all P- and R-strategies $\beta \subset \alpha$ (since α won't define n_α until all the n_β's are defined). Then, take outcome w_1.

Case 3: If α took the w_1 outcome at the end of the previous α-stage, check if there is an extension τ of $G[s-1] \upharpoonright n_\alpha$ such that $\tau \in W_j[s]$. If not, take the w_1 outcome. If such a τ is found, define $G[s] = \tau^\frown 0^\omega$ and take outcome w_2.

Case 4: If α took the w_2 outcome the last time it was eligible to act and has not been initialized, take outcome s.

Case 5: If α took the s outcome the last time it was eligible to act and has not been initialized, continue taking outcome s.

3.4. P_e-strategies and outcomes.

Let α be a P_e-strategy eligible to act at stage s.

Case 1: If α is first eligible to act at stage s or has been initialized, it defines $m_\alpha = \max\{n_\beta : n_\beta$ defined for $\beta \subset \alpha\}$ only when n_β has been defined for all $\beta \subset \alpha$. Once m_α has been defined, α waits to see if f_γ^G converges on the $2m_\alpha$th and $(2m_\alpha + 1)$st components of $\mathcal{A}$ for all S-strategies $\gamma^\frown \langle \infty \rangle \subseteq \alpha$. Until f_γ^G converges on all of those components, α remains in *Case 1* by taking the w_0 outcome and does not define define its parameter n_α.

Case 2: If α took the w_0 outcome at the previous α-stage and all maps f_γ^G have converged on the $2m_\alpha$th and $(2m_\alpha + 1)$st components of $\mathcal{A}$ for S-strategies $\gamma^\frown \langle \infty \rangle \subseteq \alpha$, it defines the parameter $n_\alpha = n$ to be large and takes outcome w_1.

Case 3: If α took the w_1 outcome at the last α-stage, check whether $\Phi_e[s]$ maps the $2n$th and $(2n + 1)$st components of $\mathcal{A}$ isomorphically into $\mathcal{B}$. If not, take outcome w_1.

If so, add a $(5n + 2)$- and $(5n + 3)$-loop to a_{2n} and a $(5n + 1)$- and $(5n + 4)$-loop to a_{2n+1} in $\mathcal{A}[s]$. Add a $(5n + 2)$- and $(5n + 4)$-loop to b_{2n} and a $(5n + 1)$- and $(5n + 3)$-loop to b_{2n+1} in $\mathcal{B}[s]$. If a map f_β^G had already been defined on these components with a use $u_{\beta,n}$ by an S-strategy β where $\beta^\frown \langle \infty \rangle \subseteq \alpha$, enumerate $u_{\beta,n}$ into G, and issue a challenge to each such S-strategy. Take outcome w_2.

Case 4: If α took the w_2 outcome at the last α-stage and has not been initialized, take outcome s.

Case 5: If α took the s outcome at the previous α-stage and has not been initialized, take outcome s again.

3.5. **S_i-strategies and outcomes.** Let α be an S_i-strategy eligible to act at stage s.

Case 1: If α is acting for the first time or has been initialized since the last α-stage, set $n_\alpha = 0$, define $f_\alpha^G[s]$ to be the empty function, and take the w_0 outcome.

Case 2: If we are not in *Case 1*, let t be the last α-stage and we break into the following subcases.

Subcase 1: α is not currently challenged by a P-strategy and one of the following conditions holds:

(1) α took the w_{n_α} outcome at stage t,
(2) α took the ∞ outcome at stage t but no P or R-strategy $\beta \supseteq \alpha^\frown \langle \infty \rangle$ took the w_1 outcome at t, or
(3) α took the ∞ outcome at t and an R-strategy $\beta \supseteq \alpha^\frown \langle \infty \rangle$ took the w_1 outcome at t but $G[s-1] \restriction (u_{\alpha,n_\alpha - 1} + 1) = G[t] \restriction (u_{\alpha,n_\alpha - 1} + 1)$.

In this case, α passes to the module for checking for extensions of f_α^G below.

Subcase 2: α is currently challenged by a P-strategy $\beta \supseteq \alpha^\frown \langle \infty \rangle$. In the verification, we will show that β is unique.

If α was challenged by β at stage t (that is, s is the first α-stage since α was challenged), then set n_α to be the least m such that $f_\alpha^G[s-1]$ is not defined on the $2m$th and $(2m+1)$st components of $\mathcal{A}$. In the verification, we will show that $n_\alpha \leq n_\beta$. Go to the module for checking for extensions of f_α^G. If f_α^G is extended to the $2n_\alpha$th and $(2n_\alpha + 1)$st components, then increment n_α. If $n_\alpha > n_\beta$, take the ∞ outcome and remove the challenge on α. Otherwise, take the w_{n_α} outcome and the challenge on α remains.

If α was challenged before stage t, then α acts in the same way except it skips the initial redefining of n_α.

Subcase 3: α is not currently challenged by a P-strategy, but α took the ∞ outcome at stage t, an R-strategy $\beta \supseteq \alpha^\frown \langle \infty \rangle$ took outcome w_1 at stage t, and $G[s-1] \restriction (u_{\alpha,n_\alpha - 1} + 1) \neq G[t] \restriction (u_{\alpha,n_\alpha - 1} + 1)$. Reset n_α to be the least m such that $f_\alpha^G[s-1]$ is not defined on the $2m$th and $(2m+1)$st components of $\mathcal{A}$. Go to the module for checking for extensions of f_α^G.

Module for checking for extensions of f_α^G: α searches for the oldest and lex-least copies of the $2n_\alpha$th and $(2n_\alpha + 1)$st components in $\mathcal{M}_i^G[s]$. If no copies are found, leave f_α^G and n_α unchanged and take outcome w_{n_α}. If α is challenged, then it remains challenged. Otherwise, if such copies are found, extend f_α^G by mapping the $2n_\alpha$th and $(2n_\alpha + 1)$st components in $\mathcal{A}$ onto their copies in $\mathcal{M}_i^G[s]$. Define u_{α,n_α} large and set it as the use of the newly defined computations of f_α^G. Increment n_α by 1 and take the ∞ outcome (unless α is challenged and we still have that $n_\alpha \leq n_\beta$, in which case, take the w_{n_α} outcome).

3.6. **Verification.** We begin with proving several auxiliary lemmas based on key observations of the construction. Afterwards, we will prove the main verification lemma. It is easy to see that the first two lemmas below are true.

Lemma 3.1. *If $f : \mathcal{A} \to \mathcal{A}$ is an embedding of $\mathcal{A}$ into itself, then f is an isomorphism.*

Lemma 3.2. *If $\mathcal{A} \cong \mathcal{M}_i^G$ for a G-computable directed graph $\mathcal{M}_i^G$ and $f^G : \mathcal{A} \to \mathcal{M}_i^G$ is an embedding defined on all of $\mathcal{A}$, then f^G is an isomorphism.*

We now prove the following lemma on challenges issued by P-strategies during the construction.

Lemma 3.3. *An S-strategy α will be challenged by at most one P-strategy at any given stage.*

Proof. Let α be an S-strategy and let β be a P-strategy such that $\beta \supseteq \alpha^\frown \langle \infty \rangle$. If β challenges α at some stage s, then it takes the w_2 outcome for the first time. The P-strategies extending $\beta^\frown \langle w_2 \rangle$ will then wait as in *Case 1*, and so will not challenge α. Until α can meet its challenge, it will continue to take the w_{n_β} outcome, and so P-strategies extending $\alpha^\frown \langle w_{n_\beta} \rangle$ will not be able to challenge α since $w_{n_\beta} \neq \infty$. $\square$

The next two lemmas will prove facts about our 1-generic G that will be useful for our S-strategies.

Lemma 3.4. *G is Δ_2^0.*

Proof. Let m be arbitrary and we will show that G can only change finitely often below m. The only strategies which can change G are the R- and P-strategies. If α is a P-strategy, then α changes G below m if and only if its parameter $n_\alpha \leq m$. If α is instead an R-strategy, then α changes G below m if and only if there is an S-strategy $\beta^\frown \langle \infty \rangle \subseteq \alpha$ such that $u_{\beta,n_\alpha} \leq m$. In either case, α will only act once to change G according to its strategy unless it is initialized and chooses new large parameters. Since parameters are always chosen large (and so are never reused), only finitely many strategies can change G below m. $\square$

Lemma 3.5. *If $\mathcal{A} \cong \mathcal{M}_i^G$, then for each n, there is an s such that for all $t \geq s$, the true copies of the $2n$th and $(2n+1)$st components of $\mathcal{A}$ in $\mathcal{M}_i^G$ are the oldest and lexicographically least isomorphic copies in $\mathcal{M}_i^G[t]$ of these components.*

Proof. By Lemma 3.4, we have that G is Δ_2^0. If $\mathcal{A} \cong \mathcal{M}_i^G$, then for each n, we have true copies of the $2n$th and $(2n+1)$st components of $\mathcal{A}$ in $\mathcal{M}_i^G$. Suppose that the associated G-uses for the true copies of the $2n$th and $(2n+1)$st components are u_{2n} and u_{2n+1}, respectively. Since G is Δ_2^0, there exists a sufficiently large stage s_n such that $G[s_n] \restriction u_{2n} = G \restriction u_{2n}$, and so after stage s_n, the copy of the $2n$th component in $\mathcal{A}$ will become the oldest and lexicographically least isomorphic copy of the component in $\mathcal{M}_i^G$. The same holds true for the $(2n+1)$st component of $\mathcal{A}$. $\square$

3.6.1. *Lemmas for each strategy.* For each strategy in our construction, we prove several key lemmas. We begin with the P- and R-strategies.

Lemma 3.6. *Let α be a P- or R-strategy. If β is another P- or R-strategy such that $\beta^\frown \langle w_0 \rangle \subseteq \alpha$, then α takes the w_0 outcome at every α-stage.*

Proof. If $\beta^\frown \langle w_0 \rangle \subseteq \alpha$ and s is an α-stage, then n_β is not defined at stage s. Therefore, α doesn't define n_α at stage s, and so it takes the w_0 outcome. $\square$

Once an R-strategy α defines its parameter, it will stay defined for the rest of the construction unless α is initialized. We show that if α is on the true path, then it will define an initial segment of G that either meets or avoids its given c.e. set. We first begin with a short lemma.

Lemma 3.7. *Let α be an R- or P-strategy that acts in Case 3 at stage s. Then $G[s] \restriction n_\alpha = G[s-1] \restriction n_\alpha$.*

Proof. If α is an R-strategy, this follows immediately because $G[s-1] \restriction n_\alpha \subseteq \tau$. If α is a P-strategy, then α may enumerate a use u_{β,n_α} into G where β is an S-strategy where $\beta^\frown\langle\infty\rangle \subseteq \alpha$. In this case, since u_{β,n_α} was chosen large, we have that $u_{\beta,n_\alpha} > n_\alpha$ and so the only change to G occurs above n_α. $\qquad\square$

Lemma 3.8. *Let α be an R_j-strategy that defines n_α at a stage s_0.*

(1) *Unless α is initialized, at all stages $s > s_0$, we have that $G[s] \restriction n_\alpha = G[s_0] \restriction n_\alpha$.*
(2) *Suppose that α acts as in Case 3 at a stage $s_1 > s_0$ with the string τ where $\tau \in W_j[s_1]$. Unless α is initialized, for all stages $s > s_1$, we have that $G[s] \restriction |\tau| = \tau$.*

Proof. Let α and n_α be as above. For (1), by Lemma 3.7 all R- or P-strategies β where $\beta \supset \alpha$ cannot change $G[s]$ below n_α, and hence cannot change G below n_α. Therefore, if there is a change to $G[s]$ below n_α, it was caused by some R- or P-strategy δ acting in *Case 3* with $\delta^\frown\langle w_1\rangle \subseteq \alpha$. But when δ acts, it takes the w_2 outcome, which initializes α.

For (2), we have that α takes the w_2 outcome at stage s_1 and the s outcome after stage s_1. All $\beta \supseteq \alpha^\frown\langle w_2\rangle$ or $\beta \supseteq \alpha^\frown\langle s\rangle$ will define large parameters $n_\beta > |\tau|$. So if there are any changes to G below $|\tau|$, it is caused by some R- or P-strategy $\delta^\frown\langle w_1\rangle \subseteq \alpha$. However, if δ changes G, then it takes the w_2 outcome, but this initializes α. $\qquad\square$

We prove something similar for the S-strategies.

Lemma 3.9. *Let α be an S-strategy and let $s_0 < s_1$ be α-stages such that $n_\alpha[s]$ has already been defined and α is not initialized between s_0 and s_1. Then*

$$G[s_0] \restriction n_\alpha[s_0] = G[s_1] \restriction n_\alpha[s_0]$$

unless some R- or P-strategy $\beta \supseteq \alpha^\frown\langle\infty\rangle$ with $n_\beta < n_\alpha[s_0]$ acts between stages s_0 and s_1.

Proof. By Lemma 3.8, G can change below $n_\alpha[s_0]$ only if an R- or P-strategy β with $n_\beta < n_\alpha[s_0]$ acts. Therefore, we cannot have that $\alpha^\frown\langle\infty\rangle <_L \beta$. We also cannot have that $\beta <_L \alpha$ or $\beta \subseteq \alpha$ since α would get initialized. Hence, it must be that $\alpha^\frown\langle\infty\rangle \subseteq \beta$. $\qquad\square$

We now prove that if a P_e-strategy α takes action to add diagonalizing loops to a pair of A-components, then the diagonalization against Φ_e on these components will remain at the end of the construction.

Lemma 3.10. *Let α be a P-strategy. Suppose that α adds diagonalizing loops to the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}$ at a stage s_0. Unless α is initialized, these components are not homogenized at a future stage.*

Proof. Let α be as in the statement of the lemma. We only homogenized these components if the path moves to the left of α, in which case α is initialized. $\qquad\square$

We now prove several key facts for the S-strategies that will help with the main verification lemma.

Lemma 3.11. *Let α be an S-strategy and β be a P-strategy such that $\alpha^\frown\langle\infty\rangle \subseteq \beta$. Suppose that there are stages $t_0 < t_1 < t_2 < t_3$ such that β defines m_β at t_0, β defines n_β at t_1, β acts in Case 3 at t_2, t_3 is the first α-stage after t_2, and β is not initialized between t_0 and t_3. Then,*

(1) *$n_\alpha[t_1] < n_\beta[t_1]$, and for all S-strategies γ such that $\gamma^\frown\langle\infty\rangle \subseteq \beta$, f_γ^G is defined on the $2m_\beta$th and $(2m_\beta+1)$st components of $\mathcal{A}$ with $n_\beta[t_1] > u_{\gamma,m_\beta}[t_1]$;*
(2) *for $t_1 \leq t < t_2$, $n_\alpha[t] > m_\beta[t] = m_\beta[t_0]$; and*

(3) $n_\alpha[t_3] > m_\beta[t_3] = m_\beta[t_0]$.

Proof. Note that the values of m_β and n_β do not change once they are defined since β is not initialized. For (1), β defines n_β large at stage t_1, and so $n_\beta[t_1] > n_\alpha[t_1]$. In addition, β does not define n_β until all S-strategies $\gamma^\frown\langle\infty\rangle \subseteq \beta$ have $n_\gamma > m_\beta$. Therefore, when n_β is chosen large at stage t_1, we have that $n_\beta[t_1] > u_{\gamma,m_\beta}[t_1]$.

For (2), when β defines n_β at stage t_1, f_α^G is defined on the $2m_\beta$th and $(2m_\beta + 1)$st components of $\mathcal{A}$. Therefore, we have that $n_\alpha[t_1] > m_\beta[t_0]$. At stage t_1, β takes the w_1 outcome for the first time, and so all strategies γ where $\beta^\frown\langle w_1\rangle \subseteq \gamma$ choose parameters larger than $n_\alpha[t_1]$. In particular, none of these strategies can lower the value of n_α below m_β. The only other strategies which can lower n_α without initializing α are γ where $\alpha^\frown\langle\infty\rangle \subseteq \gamma \subset \beta$. However, these γ would initialize β. Hence, we have that for all $t_1 \leq t < t_2$ that $n_\alpha[t] > m_\beta$. In addition, at stage t_2, β acts after α, so n_α does not get redefined to reflect β's action until the α-stage t_3. It follows that $n_\alpha[t] > m_\beta$ for $t_2 \leq t < t_3$.

For (3), at stage t_3, α sets $n_\alpha[t_3]$ to be the least m such that $f_\alpha^G[t_3]$ is not defined on the $2m$th and $(2m+1)$st components of $\mathcal{A}$. At stage t_2, β enumerated u_{γ,n_β} for S-strategies $\gamma^\frown\langle\infty\rangle \subseteq \beta$. However, $u_{\gamma,n_\beta} > n_\beta$ because u_{γ,n_β} was defined large when it was chosen and $n_\beta > u_{\alpha,m_\beta}$ by (1). Therefore, the computation f_α^G on the $2m_\beta$th and $(2m_\beta+1)$st components is not destroyed by β's action at stage t_2. It follows that $n_\alpha[t_3] > m_\beta[t_3] = m_\beta[t_0]$. $\qquad\square$

Lemma 3.12. *Let α be an S-strategy and β be an R-strategy such that $\alpha^\frown\langle\infty\rangle \subseteq \beta$. Suppose that there are stages $t_0 < t_1 < t_2 < t_3$ such that β defines m_β at t_0, β defines n_β at t_1, β acts in Case 3 at t_2, t_3 is the next α-stage after t_2, and β is not initialized between t_0 and t_3. Then,*

(1) $m_\beta < n_\alpha[t_1] < n_\beta[t_1]$ and $n_\beta[t_1] > u_{\alpha,m_\beta}[t_1]$, which is the use of $f_\alpha^G[t_1]$ on the $2m_\beta$th and $(2m_\beta + 1)$st components of $\mathcal{A}$;
(2) for all $t_1 \leq t < t_3$, $n_\alpha[t] > m_\beta$; and
(3) $n_\alpha[t_3] > m_\beta$.

Proof. (1) and (2) hold as in Lemma 3.11. For (3), when β acts at stage t_2, it defines $G[t_2] = \tau^\frown 0^\omega$ where $G[t_2 - 1] \upharpoonright n_\beta \subseteq \tau$. Therefore, if $u_{\alpha,m_\beta}[t_1]$ is the use of the map $f_\alpha^G[t_1]$ on the $2m_\beta$th and $(2m_\beta + 1)$st components, then

$$G[t_2] \upharpoonright u_{\alpha,m_\beta} = G[t_2 - 1] \upharpoonright u_{\alpha,m_\beta}.$$

Moreover, no requirements can change G below n_β between stages t_2 and t_3 without initializing β. Hence,

$$G[t_3 - 1] \upharpoonright u_{\alpha,m_\beta} = G[t_2 - 1] \upharpoonright u_{\alpha,m_\beta}.$$

In particular, when α acts in *Case 2* at stage t_3, either α acts in *Subcase 1* and doesn't change n_α, or it acts in *Subcase 3* and we retain $n_\alpha[s_3] > m_\beta$ because G has not changed below u_{α,m_β}. $\qquad\square$

We now combine Lemmas 3.11 and 3.12 into the following single lemma.

Lemma 3.13. *Let α be an S-strategy. An R- or P-strategy $\beta \supseteq \alpha^\frown\langle\infty\rangle$ cannot cause n_α to drop below m_β.*

With the help of Lemma 3.13, we now prove the following important fact about the S-strategies.

Lemma 3.14. *Let α be an S-strategy that is never initialized after stage s and takes the ∞ outcome infinitely often. For all m, there exists a stage t_m such that $n_\alpha[t] \geq m$ for all $t \geq t_m$.*

Proof. Fix m. Let $\beta_0, \ldots, \beta_k$ be the R- and P-strategies such that $\alpha^\frown \langle \infty \rangle \subseteq \beta_i$ and m_{β_i} is defined at some stage where $m_{\beta_i} < m$. Let $t > s$ be a stage such that no β_i acts as in *Case 3* to change G with $m_{\beta_i} < m$ after stage t. After stage t, no requirement can cause n_α to drop below m. Moreover, no other R- or P-requirement δ will act until $n_\alpha > m_\delta > m$. Let $t_m > t$ be the least stage such that $n_\alpha[t_m] > m$. $\square$

3.6.2. *Lemmas on interactions between multiple strategies.* In this section, we prove several lemmas that detail how tension between multiple strategies are resolved in our construction. We first show that for a lower priority R-strategy, it won't be able to injure higher priority S-strategies at arbitrarily small numbers.

Lemma 3.15. *Let α be an R_j-strategy that acts in Case 3 at stage s. For all P- and R-strategies $\beta \subset \alpha$ and all S-strategies γ such that $\gamma^\frown \langle \infty \rangle \subseteq \alpha$, if f_γ^G is defined on the $2n_\beta$th and $(2n_\beta + 1)$st components of $\mathcal{A}$, then*

$$G[s] \upharpoonright u_{\gamma, n_\beta} = G[s-1] \upharpoonright u_{\gamma, n_\beta}.$$

Therefore, these maps remain defined.

Proof. When α first acts, it waits for n_β to be defined for all $\beta \subset \alpha$ and then defines $m_\alpha = \max\{n_\beta : \beta \subset \alpha\}$ and does not define n_α until it sees that for all S-strategies $\gamma^\frown \langle \infty \rangle \subseteq \alpha$, the map f_γ^G has been defined on the $2m_\alpha$th and $(2m_\alpha + 1)$st components of $\mathcal{A}$. When α is finally able to define n_α, it picks n_α large and so $n_\alpha > u_{\gamma, n_\beta}$ for all S-strategies $\gamma^\frown \langle \infty \rangle \subseteq \alpha$ and for all P- and R-strategies $\beta \subset \alpha$. If α finds a τ such that $\tau \in W_j[s]$ and $G[s-1] \upharpoonright n_\alpha \subseteq \tau$ at a stage s, it sets $G[s] = \tau^\frown 0^\omega$. This will not change G below u_{γ, n_β} for all γ and β as above since $n_\alpha > u_{\gamma, n_\beta}$. $\square$

The following lemma details how the S-strategies are able to undo any of their previously defined maps on $\mathcal{A}$-components in the event that a pair of components were used to diagonalize against a computable Φ_e by some P_e-strategy.

Lemma 3.16. *Let α be an S_i-strategy. If $f_\alpha^G[s]$ is defined on the $2m$th and $(2m+1)$st components of $\mathcal{A}$ at stage s, then $l_m[s] < u_{\alpha, m}[s]$ where $l_m[s]$ is the max use of the edges in the image of these components in $\mathcal{M}_i^G$ and $u_{\alpha, m}[s]$ is the use of this computation. In particular, any change in these loops in $\mathcal{M}_i^G$ causes the computation to be destroyed.*

Proof. Whenever an $f_\alpha^G[s]$ computation is defined, its use is defined large. Therefore, we have that $l_m[s] < u_{\alpha, m}[s]$ when the computation is first defined. The computation may be destroyed later by a change in G below $u_{\alpha, m}[s]$. However, if it is later restored by $u_{\alpha, m}[t] = u_{\alpha, m}[s]$ for $t > s$, then $l_m[t] = l_m[s]$ and so the loops in $\mathcal{M}_i^G$ are restored as well. $\square$

We now show that enumerating these uses has the intended effect of deleting a map f_α^G defined previously by some S-strategy α on some $\mathcal{A}$-components.

Lemma 3.17. *Let α be a P-strategy and β be an S-strategy such that $\beta^\frown \langle \infty \rangle \subseteq \alpha$. Suppose $s_0 < s_1 < s_2$ are stages such that α defines n_α at s_0, β defined f_β^G on the $2n_\alpha$th and $(2n_\alpha + 1)$st components in $\mathcal{A}$ at s_1, α acts in Case 3 at s_2 and puts u_{β, n_α} into $G[s_2]$, and α is not initialized between s_0 and s_2. Then,*

(1) $n_\alpha[s_0] > n_\beta[s_0]$,

(2) *for all $s_1 \leq t < s_2$, we have that the values of $n_\alpha[t] = n_\alpha[s_0]$ and $u_{\beta,n_\alpha}[t] = u_{\beta,n_\alpha}[s_1]$, so we denote them by n_α and u_{β,n_α},*

(3) *for all $t < s_2$, $u_{\beta,n_\alpha} \notin G[t]$ and therefore $G[s_2] \restriction (u_{\beta,n_\alpha}+1) \neq G[t] \restriction (u_{\beta,n_\alpha}+1)$ for all $t < s_2$,*

(4) *unless α is initialized, $u_{\beta,n_\alpha} \in G[s]$ for all $s \geq s_2$, and*

(5) *if there is an $s \geq s_2$ such that $u_{\beta,n_\alpha} \notin G[s]$, then the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}$ and $\mathcal{B}$ are homogenized.*

Proof. (1) follows from the fact that n_α is defined large at stage s_0. For (2), n_α changes values only when α is initialized, and the only strategies that can change G below u_{β,n_α} would initialize α. For (3), when β defines u_{β,n_α} at stage s_1, the use was chosen large so $u_{\beta,n_\alpha} \notin G[t]$ for $t \leq s_1$. For $s_1 < t < s_2$, the only strategies that could change G below u_{β,n_α} would initialize α by Lemma 3.15. Lastly for (4), only an R-strategy γ could remove u_{β,n_α} and only if $n_\gamma < u_{\beta,n_\alpha}$. But such an R-strategy with $n_\gamma < u_{\beta,n_\alpha}$ would initialize α when it acts by Lemma 3.15. Furthermore, the path $\pi_s <_L \alpha$ causes the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}$ and $\mathcal{B}$ to be homogenized, proving (5). $\square$

Lemma 3.18. *Let α be a P-strategy and β be an S-strategy such that $\beta^\frown\langle\infty\rangle \subseteq \alpha$. Suppose α acts at stage t in Case 3 and puts u_{β,n_α} into G and challenges β. Unless β is initialized, at the next β-stage s, β defines $n_\beta[s] \leq n_\alpha[s] = n_\alpha[t]$.*

Proof. By Lemma 3.17 and the fact that β is not initialized, we get that $u_{\beta,n_\alpha} \in G[s-1]$. Therefore, $f_\beta^G[s-1]$ is no longer defined on the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}$ and hence n_β is redefined to a value which is at most n_α. $\square$

We now state and prove the main verification lemma for our construction.

Lemma 3.19 (Main Verification Lemma). *Let $\pi = \liminf_s \pi_s$ be the true path of the construction, where π_s denotes the current true path at stage s of the construction. Let $\alpha \subset \pi$.*

(1) *If α is an R_j-strategy, then the parameters m_α and n_α are eventually permanently defined and there is an outcome o and an α-stage t_α such that for all α-stages $s \geq t_\alpha$, α takes outcome o where o ranges over $\{s, w_1\}$.*

(2) *Let α be a P_e-strategy, then the parameters m_α and n_α are eventually permanently defined and there is an outcome o and an α-stage t_α such that for all α-stages $s \geq t_\alpha$, α takes outcome o where o ranges over $\{s, w_1\}$.*

(3) *If α is an S_i-strategy, then either α takes outcome ∞ infinitely often or there is an outcome w_n and a stage t_α such that for all α-stages $s > t_\alpha$, α takes outcome w_n. If $\mathcal{A} \cong \mathcal{M}_i^G$, then α takes the ∞ outcome infinitely often and defines an embedding $f_\alpha^G : \mathcal{A} \to \mathcal{M}_i^G$ which can be extended in a G-computable way to a G-computable isomorphism $\hat{f}_\alpha^G$ between $\mathcal{A}$ and $\mathcal{M}_i^G$.*

In addition, α satisfies its assigned requirement.

Proof. For (1), let $\alpha \subset \pi$ be an R_j-strategy and let s_0 be the least stage such that $\alpha \leq_L \pi_s$ for all $s \geq s_0$ and for all R- and P-strategies $\beta \subset \alpha$, n_β is defined. At stage s_0, α defines m_α permanently since α is never initialized again. By Lemma 3.14, there is an $s_1 > s_0$ such that for all S-strategies β where $\beta^\frown\langle\infty\rangle \subseteq \alpha$, we have $n_\beta[t] > m_\alpha$ for all $t \geq s_1$. At stage s_1 (if not before), α defines n_α permanently and takes outcome w_1.

If α remains in the first part of *Case 3* from its description for all α-stages $s \geq s_0$, then R_j is satisfied because for all extensions τ of $\sigma = G[s_0] \restriction n_\alpha$, $\tau \notin W_j$. Moreover, since $\alpha \subset \pi$,

we have that $G[s_0] \restriction n_\alpha = G \restriction n_\alpha$ by Lemma 3.8. Otherwise, there is a stage $s \geq s_0$ such that α finds an extension $\tau \supseteq G[s_0] \restriction n_\alpha$ where $\tau \in W_j[s]$. At the end of stage s, α takes the w_2 outcome, and since $\alpha \subset \pi$, α will be able to act again and finally take the s outcome. For the rest of the construction, it will be in *Case 5* of its description after defining $\tau \subseteq G$ where $\tau \in W_j$. Again by Lemma 3.8, we have that $G[t] \restriction |\tau| = \tau$ for all $t \geq s$, and R_j is satisfied.

We now prove (2). Suppose $\alpha \subset \pi$ is a P_e-strategy. Let s_0 be the least stage such that $\alpha \leq_L \pi_s$ for all $s \geq s_0$ and for all R- or P-strategies $\beta \subset \alpha$, n_β is defined. Like before, at stage s_0, α defines m_α permanently since α is never initialized. By Lemma 3.14, there exists a stage $s_1 > s_0$ such that for all S-strategies β with $\beta^\frown \langle \infty \rangle \subseteq \alpha$, we have that $n_\beta[t] > m_\alpha$ for all $t \geq s_1$. Then at stage s_1 (if not before), α defines n_α permanently and takes outcome w_1.

If α remains in the first part of *Case 3* from its description for all α-stages $s \geq s_0$, then P_e is trivially satisfied since Φ_e is not total or maps $\mathcal{A}$ incorrectly into $\mathcal{B}$, and so it cannot be an isomorphism between $\mathcal{A}$ and $\mathcal{B}$. In this case, α takes outcome w_1 cofinitely often.

Otherwise, there is an α-stage $s_1 > s_0$ where $\Phi_e[s_1]$ maps the $2n$th and $(2n+1)$st components of $\mathcal{A}$ isomorphically into $\mathcal{B}$. Then, α carries out all actions described in the second part of *Case 3*. Let $s_2 > s_1$ be the next α-stage, and now α is in *Case 4* of its description and can now take the s outcome. Moreover, since $\alpha \subset \pi$, α will never get initialized again and so the $2n$th and $(2n+1)$st components of $\mathcal{A}$ and $\mathcal{B}$ are never homogenized. By Lemma 3.10 we have that $\Phi_e(a_{2n}) = b_{2n}$, but a_{2n} is connected to a cycle of length $5n+3$ whereas b_{2n} is connected to a cycle of length $5n+4$. Similarly, $\Phi_e(a_{2n+1}) = b_{2n+1}$, but a_{2n+1} is connected to a cycle of length $5n+4$ whereas b_{2n+1} is connected to a cycle of length $5n+3$. Hence, Φ_e cannot be an isomorphism between $\mathcal{A}$ and $\mathcal{B}$, and so P_e is satisfied.

For (3), let $\alpha \subset \pi$ be an S_i-strategy and let s_0 be the least stage such that for all stages $s \geq s_0$, $\alpha \leq_L \pi_s$. Under all of the subcases of *Case 3*, α enacts the module to search for extensions of f_α^G on the $2n_\alpha$th and $(2n_\alpha + 1)$st components of $\mathcal{A}$ for its currently defined n_α parameter. If an extension cannot be found for these components, then we have that α takes the w_{n_α} outcome at cofinitely many α-stages after s_0 and we are done since $\mathcal{A} \not\cong \mathcal{M}_i^G$.

We can then assume that $\mathcal{A} \cong \mathcal{M}_i^G$ and that α takes the ∞ outcome infinitely often. α can then eventually find a correct extension on these components by Lemma 3.5. Additionally, we have by Lemma 3.14 that $n_\alpha[s] \to \infty$ as $s \to \infty$ where $n_\alpha[s]$ denotes the value of n_α at the end of stage s. We also have that by Lemmas 3.16 and 3.18, these embeddings will remain or be able to recover if they are challenged by lower priority strategies.

Although f_α^G is not defined on the homogenizing loops added at the end of each stage, we can G-computably extend f_α^G to a map $\hat{f}_\alpha^G$ defined on all of $\mathcal{A}$ in the following way. Since $\mathcal{A} \cong \mathcal{M}_i^G$, then these loops added at the end of each stage will eventually have true copies in the images under f_α^G on the affected components, and they will become the oldest and lex-least such copies by Lemma 3.5. We can find copies of these new loops by using G since $\mathcal{M}_i^G$ is G-computable, and once we know when these loops appear, we can extend f_α^G appropriately on these new components to obtain $\hat{f}_\alpha^G$. Our new map $\hat{f}_\alpha^G$ is still G-computable, and by Lemma 3.2, $\hat{f}_\alpha^G$ is our G-computable isomorphism between $\mathcal{A}$ and $\mathcal{M}_i^G$. $\square$

4. OTHER CLASSES OF STRUCTURES

For this section, we restate the main result from [Vil25].

Theorem 4.1. *Let $P = (P, \leq)$ be a computable partially ordered set and let $P = P_0 \sqcup P_1$ be a computable partition. Then, there exists a computable computably categorical directed graph $\mathcal{G}$ and an embedding h of P into the c.e. degrees where $\mathcal{G}$ is computably categorical relative to each degree in $h(P_0)$ and is not computably categorical relative to each degree in $h(P_1)$.*

We now show that the structure which witnesses the above behavior need not be a directed graph.

Theorem 4.2. *Let $P = (P, \leq)$ be a computable partially ordered set and let $P = P_0 \sqcup P_1$ be a computable partition. For the following classes of structures, there exists a computable example in each class which satisfies the conclusion of Theorem 4.1: symmetric, irreflexive graphs; partial orderings; lattices; rings with zero-divisors; integral domains of arbitrary characteristic; commutative semigroups; and 2-step nilpotent groups.*

We will use the codings given by the authors in [Hir+02] in order to code the directed graph $\mathcal{G}$ (and presentations of it) in the statement of Theorem 4.1 into a structure in one of the above classes of structures.

4.1. **Codings.** Suppose an abstract graph $\mathcal{G}$ and a structure $\mathcal{A}$, from one of the listed classes in the statement of Theorem 4.2, have computable presentations. Let G and A be particular copies of $\mathcal{G}$ and $\mathcal{A}$, respectively. We first state the following definitions before defining the coding from [Hir+02].

Definition 4.3. A relation U on a structure $\mathcal{A}$ is *invariant* if for every automorphism $f : \mathcal{A} \cong \mathcal{A}$, we have that $f(U) = U$.

Here, an n-ary relation U on a structure $\mathcal{A}$ is some subset of $|\mathcal{A}|^n$ where $|\mathcal{A}|$ denotes the underlying domain of $\mathcal{A}$.

Definition 4.4. A relation U on the domain of a structure $\mathcal{A}$ is *intrinsically computable* if for any computable $\mathcal{B}$ and computable isomorphism $f : \mathcal{A} \to \mathcal{B}$, the image $f(U)$ is computable.

Definition 4.5. Let $\mathbf{d}$ be a degree. A $\mathbf{d}$-*computable defining family* for a structure $\mathcal{A}$ is a $\mathbf{d}$-computable set of existential formulas $\varphi_0(\vec{a}, x), \varphi_1(\vec{a}, x), \ldots$ such that $\vec{a}$ is a tuple of elements of $|\mathcal{A}|$, each $x \in |\mathcal{A}|$ satisfies some φ_i, and no two elements of $|\mathcal{A}|$ satisfy the same φ_i.

The main idea of the coding methods given in [Hir+02] is that there are intrinsically computable, invariant relations $D(x)$ and $R(x, y)$ on the domain of $\mathcal{A}$ such that taking the elements in $|\mathcal{A}|$ satisfying $D(x)$ and adding the relation on them defined by $R(x, y)$ gives a copy of the graph $\mathcal{G}$. This gives us a map from copies of the structure $\mathcal{A}$ to copies of $\mathcal{G}$, which we will write as $A \mapsto G_A$. In addition, there is a uniform computable functional taking copies of $\mathcal{G}$ to copies of $\mathcal{A}$, which we will write as $G \mapsto A_G$. Note that these maps can be applied repeatedly. For example, we can apply a map to go from A to G_A and then to A_{G_A}, a copy of the structure $\mathcal{A}$. Note that A_{G_A} and A are isomorphic as both are copies of the structure $\mathcal{A}$, but they are *not* the same presentation.

The coding methods in [Hir+02] satisfy the following list of properties:

(P0) For every presentation G of $\mathcal{G}$, the structure A_G is $\deg(G)$-computable.

(P1) For every presentation G of $\mathcal{G}$, there is a $\deg(G)$-computable map $g_G : |A_G| \to G$ such that $R^{A_G}(x, y) \iff E^G(g_G(x), g_G(y))$ for all $x, y \in |A_G|$.

(P2) If $f : |\mathcal{A}| \to |\mathcal{A}|$ is 1-to-1 and onto and $R(x, y) \iff R(f(x), f(y))$ for all $x, y \in |\mathcal{A}|$, then f can be extended to an automorphism of $\mathcal{A}$.

(P3) For every presentation G of $\mathcal{G}$, there is a $\deg(G)$-computable set of existential formulas $\varphi_0(\vec{a}, \vec{b}_0, x), \varphi_1(\vec{a}, \vec{b}_1, x), \ldots$ such that $\vec{a}$ is a tuple of elements from the universe of A_G, each $\vec{b}_i$ is a tuple of elements of $|A_G|$, each x in the universe of A_G satisfies some φ_i, and no two elements of the universe of A_G satisfy the same φ_i.

Each of these properties are key in proving Lemmas 2.6-2.9 in [Hir+02], whose relativized versions below are needed to prove Theorem 4.2. Here, we let $\mathcal{G}$ be the directed graph that satisfies the conclusion of Theorem 4.1, but the lemmas hold in general for any directed graph and for all degrees $\mathbf{d}$.

Lemma 4.6. *For every $\mathbf{d}$-computable presentation G of $\mathcal{G}$, there is a $\mathbf{d}$-computable isomorphism from G_{A_G} onto G.*

Lemma 4.7. *For every $\mathbf{d}$-computable presentation A of $\mathcal{A}$, there is a $\mathbf{d}$-computable isomorphism from A_{G_A} onto A.*

Lemma 4.8. *If G_0 and G_1 are $\mathbf{d}$-computable presentations of $\mathcal{G}$ and $h : G_0 \to G_1$ is a $\mathbf{d}$-computable isomorphism, then there is a $\mathbf{d}$-computable isomorphism $\hat{h} : A_{G_0} \to A_{G_1}$.*

Lemma 4.9. *If A_0 and A_1 are $\mathbf{d}$-computable presentations of $\mathcal{A}$ and $h : A_0 \to A_1$ is a $\mathbf{d}$-computable isomorphism, then there is a $\mathbf{d}$-computable isomorphism $\hat{h} : G_{A_0} \to G_{A_1}$.*

We now prove a key lemma with the help of these relativized lemmas.

Lemma 4.10. *For any degree $\mathbf{d}$, $\mathcal{G}$ is computably categorical relative to $\mathbf{d}$ if and only if $\mathcal{A}$ is computably categorical relative to $\mathbf{d}$.*

Proof. Fix the degree $\mathbf{d}$. Suppose $\mathcal{G}$ is computably categorical relative to $\mathbf{d}$. Let A_0 and A_1 be $\mathbf{d}$-computable presentations of $\mathcal{A}$. We then obtain $\mathbf{d}$-computable presentations G_{A_0} and G_{A_1} of $\mathcal{G}$. Since $\mathcal{G}$ is computably categorical relative to $\mathbf{d}$, there is a $\mathbf{d}$-computable isomorphism $h : G_{A_0} \to G_{A_1}$. By Lemma 4.8, we have a $\mathbf{d}$-computable isomorphism $\hat{h} : A_{G_{A_0}} \to A_{G_{A_1}}$. Additionally, by Lemma 4.7, there are $\mathbf{d}$-computable isomorphisms $f_0 : A_{G_{A_0}} \to A_0$ and $f_1 : A_{G_{A_1}} \to A_1$. It follows that $f_1 \circ \hat{h} \circ f_0^{-1} : A_0 \to A_1$ is a $\mathbf{d}$-computable isomorphism as needed.

For the reverse direction, suppose $\mathcal{A}$ is computably categorical relative to $\mathbf{d}$. To show $\mathcal{G}$ is computably categorical relative to $\mathbf{d}$, we run an identical argument as above using Lemmas 4.6 and 4.9 in place of Lemmas 4.7 and 4.8, respectively. $\qquad\square$

Theorem 4.2 follows almost immediately.

Proof of Theorem 4.2. Let P be our computable poset where $P = P_0 \sqcup P_1$ is a computable partition, let $\mathcal{G}$ be the computable directed graph which witnesses Theorem 4.1, and let h be the embedding of P into the c.e. degrees. By Lemma 4.10, we have that since $\mathcal{G}$ is computably categorical, so is $\mathcal{A}$. We also have that because $\mathcal{G}$ is computably categorical relative to all $\mathbf{d} \in h(P_0)$ that $\mathcal{A}$ is computably categorical relative to all $\mathbf{d} \in h(P_0)$. Finally, since $\mathcal{G}$ is not computably categorical relative to any $\mathbf{d} \in h(P_1)$, it follows that $\mathcal{A}$ is also not computably categorical relative to any $\mathbf{d} \in h(P_1)$. $\qquad\square$

4.2. **Linear orderings and Boolean algebras.** To end this section, we discuss two classes of structures in which the version of Theorem 4.2 where the structure is not computably categorical fails. That is, in both classes of structures, it is impossible to produce a computable structure $\mathcal{A}$ which is not computably categorical and a c.e. degree $\mathbf{d}$ such that $\mathcal{A}$ is computably categorical relative to $\mathbf{d}$. These observations illustrate the limitations of the techniques from [Vil25] as a result of key structural facts about certain classes of structures, and not from the properties of certain degrees as with 2-generics.

For computable linear orderings, Remmel in [Rem81] proved the following.

Theorem 4.11 (Remmel [Rem81]). *A computable linear order is computably categorical if and only if it has finitely many adjacent pairs.*

Here, we say that for a linear ordering $(L, \leq)$, two elements x, y are an *adjacent pair* if either $x < y$ or $y < x$ and there is no $z \in L$ such that $x < z < y$ or $y < z < x$. Remmel's result also fully characterizes relative computable categoricity for computable linear orderings. So, if L is a computable linear ordering which is computably categorical, then it must be computably categorical relative to all degrees $\mathbf{d}$, and so we cannot create a c.e. degree $\mathbf{d}$ such that L is not computably categorical relative to $\mathbf{d}$.

Remmel's construction in the only if direction of his result also relativizes to any set, so for linear orderings which are not computably categorical, they must also fail to be categorical relative to all degrees $\mathbf{d}$. This differs greatly from what happens in the classes of structures listed in Theorem 4.2, where computable examples could fail to be computably categorical but still be categorical relative to some c.e. degrees.

We now consider the class of computable Boolean algebras. For these structures, we have the following results.

Theorem 4.12 (Goncharov [Gon75]). *A computable Boolean algebra is computably categorical if and only if it has finitely many atoms.*

This fully characterizes relative computable categoricity for computable Boolean algebras. Like with computable linear orderings, if a computable Boolean algebra $\mathcal{B}$ is computably categorical, then it must be computably categorical relative to all degrees $\mathbf{d}$. That is, no computable computably categorical Boolean algebra can witness the nonmonotonic behavior of computably categoricity relative to a degree. Additionally, Bazhenov showed the following.

Theorem 4.13 (Bazhenov [Baz14]). *For every degree $\mathbf{d} < \mathbf{0}'$, a computable Boolean algebra is $\mathbf{d}$-computably categorical if and only if it is computably categorical.*

By Bazhenov's result, if $\mathcal{B}$ is not computably categorical, then for all degrees $\mathbf{d} < \mathbf{0}'$, $\mathcal{B}$ is not $\mathbf{d}$-computably categorical. This implies that $\mathcal{B}$ is not computably categorical relative to any $\mathbf{d} < \mathbf{0}'$, and so Theorem 4.2 also fails for the class of computable Boolean algebras.

In light of the observations made above for linear orderings and Boolean algebras, we pose the following question.

Question 4.14. *For other classes of structures with an algebraic characterization of relative computable categoricity, does there exist an example of a structure which is not computably categorical but is computably categorical relative to some degree $\mathbf{d} > \mathbf{0}$?*

The list of structures in this section is not exhaustive, and there are several examples in the literature in which computable categoricity and relativized computable categoricity coincide. One such example is the class of trees, in which all trees with finite height are computably

categorical and relatively computably categorical [Lem+05]. For those classes of structures, it would be interesting to know if their limiting behavior in regards to categoricity relative to a degree already occurs at the level of **0**.

References

[AC16]   B. Anderson and B. Csima. "Degrees That Are Not Degrees of Categoricity". *Notre Dame Journal of Formal Logic* 57.3 (2016), 389 –398.

[Ash00]   C. J. Ash. *Computable Structures and the Hyperarithmetical Hierarchy.* Ed. by J. Knight. New York: Elsevier, 2000.

[Ash+89]   C. Ash et al. "Generic copies of countable structures". *Annals of Pure and Applied Logic* 42.3 (1989), 195–205. ISSN: 0168-0072.

[Baz14]   N. A. Bazhenov. "Δ_2^0-Categoricity of Boolean Algebras". *Journal of Mathematical Sciences* 203.4 (2014), 444–454.

[Chi90]   J. Chisholm. "Effective Model Theory vs. Recursive Model Theory". *The Journal of Symbolic Logic* 55.3 (1990), 1168–1191. ISSN: 00224812.

[DHTM21]   R. Downey, M. Harrison-Trainor, and A. Melnikov. "Relativizing computable categoricity". *Proc. Amer. Math. Soc.* 149.9 (2021), 3999–4013. ISSN: 0002-9939.

[Fra17]   J. N. Y. Franklin. "Strength and Weakness in Computable Structure Theory". *Computability and Complexity: Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday.* Ed. by A. Day et al. Cham: Springer International Publishing, 2017, 302–323. ISBN: 978-3-319-50062-1.

[FS14]   J. N. Y. Franklin and R. Solomon. "Degrees that Are Low for Isomorphism". *Computability* 3 (2014), 73–89.

[Gon75]   S. S. Goncharov. "Some properties of the constructivization of Boolean algebras". *Siberian Mathematical Journal* 16.2 (1975), 203–214.

[Gon80]   S. S. Goncharov. "The problem of the number of nonautoequivalent constructivizations". *Algebra i Logika* 19.6 (1980), 621–639, 745.

[Hir+02]   D. R. Hirschfeldt et al. "Degree spectra and computable dimensions in algebraic structures". *Annals of Pure and Applied Logic* 115.1 (2002), 71–113. ISSN: 0168-0072.

[Lem+05]   S. Lempp et al. "Computable categoricity of trees of finite height". *Journal of Symbolic Logic* 70.1 (2005), 151 –215.

[Mon21]   A. Montalbán. *Computable Structure Theory: Within the Arithmetic.* Perspectives in Logic. Cambridge University Press, 2021.

[Rem81]   J. B. Remmel. "Recursively Categorical Linear Orderings". *Proc. Amer. Math. Soc.* 83.2 (1981), 387–391.

[Vil25]   J. D. Villano. "Computable categoricity relative to a c.e. degree". *Notre Dame Journal of Formal Logic* (2025). To appear.

Department of Mathematics, University of Connecticut, Storrs, Connecticut 06269
Email address: javavill@uconn.edu