DEGREES OF CATEGORICITY AND THE HYPERARITHMETIC HIERARCHY

BARBARA F. CSIMA, JOHANNA N. Y. FRANKLIN, AND RICHARD A. SHORE

Abstract. We study arithmetic and hyperarithmetic degrees of categoricity. We extend a result of Fokina, Kalimullin, and R. Miller to show that for every computable ordinal α, $\mathbf{0}^{(\alpha)}$ is the degree of categoricity of some computable structure $\mathcal{A}$. We show additionally that for α a computable successor ordinal, every degree 2-c.e. in and above $\mathbf{0}^{(\alpha)}$ is a degree of categoricity. We further prove that every degree of categoricity is hyperarithmetic and show that the index set of structures with degrees of categoricity is Π^1_1 complete.

1. Introduction

Though classically two structures are considered the same if they are isomorphic, in computable model theory we must distinguish between particular presentations of a structure, since they may have different computable properties. We say a structure is computably categorical, if between any two computable presentations that are isomorphic, there exists a computable isomorphism. For example, the structure $(\eta, <)$, the countable dense linear order without endpoints, is easily seen to be computably categorical since the usual back-and-forth isomorphism can be constructed computably between computable copies. However, there are many computable structures that are not computably categorical. A well-known example is the structure $(\omega, <)$. It is easy to construct a computable copy, $\widehat{\mathcal{N}}$, of $(\omega, <)$ such that any isomorphism between it and the standard copy, $\mathcal{N}$, computes the halting set. It is also easy to see that in order to construct an isomorphism between two copies of $(\omega, <)$, all we need to do is identify the least element of each

Date: August 18, 2011.

B. Csima was partially supported by Canadian NSERC Discovery Grant 312501. R. Shore was partially supported by NSF Grant DMS-0852811 and John Templeton Foundation Grant 13408. This project was started by B. Csima and R. Shore while the former was visiting MIT, and we thank them for their hospitality. B. Csima would also like to thank the Max Planck Institute for Mathematics in Bonn, Germany for a productive visit.

order, then the next element of each order, and so on. These are Σ^0_1 questions, so $\mathbf{0}'$ can certainly build an isomorphism. This leads us to recall the relativized definition of computable categoricity.

Definition 1.1. A computable structure $\mathcal{C}$ is $\mathbf{d}$-*computably categorical* for a Turing degree $\mathbf{d}$ if, for every computable $\mathcal{A} \cong \mathcal{C}$, there is a $\mathbf{d}$-computable isomorphism from $\mathcal{C}$ to $\mathcal{A}$.

We have observed above that the structure $(\omega, <)$ is not only $\mathbf{0}'$-computably categorical, but that if $\mathbf{c}$ is such that $(\omega, <)$ is $\mathbf{c}$-computably categorical, then $\mathbf{d} \geq \mathbf{0}'$ (since, in particular, $\mathbf{c}$ computes an isomorphism between $\mathcal{N}$ and $\widehat{\mathcal{N}}$). Thus, for the structure $(\omega, <)$, the degree $\mathbf{0}'$ exactly describes the difficulty of computing isomorphisms between copies of the structure.

This natural idea of a "degree of categoricity" for a computable structure was first introduced by Fokina, Kalimullin and Miller in [2].

Definition 1.2. [2] A Turing degree $\mathbf{d}$ is said to be the degree of categoricity of a computable structure $\mathcal{C}$ if $\mathbf{d}$ is the least degree such that $\mathcal{C}$ is $\mathbf{d}$-computably categorical. The degree $\mathbf{d}$ is a *degree of categoricity* if it is the degree of categoricity of some computable structure.

So far, every known degree of categoricity $\mathbf{d}$ has the following additional property: There is a structure $\mathcal{C}$ with computable copies $\mathcal{C}_1$ and $\mathcal{C}_2$ such that not only does $\mathcal{C}$ have degree of categoricity $\mathbf{d}$, but every isomorphism $f : \mathcal{C}_1 \cong \mathcal{C}_2$ computes $\mathbf{d}$. If a degree of categoricity has this property, we say it is a *strong degree of categoricity.*

In [2], Fokina, Kalimullin, and Miller proved that for all $m < \omega$, $\mathbf{0}^{(m)}$ (and, in fact, any degree 2-c.e. in and above $\mathbf{0}^{(m)}$) is a strong degree of categoricity, as is $\mathbf{0}^{(\omega)}$. They prove the results first for $m = 1$ and then use relativization and Marker's extensions [5] to transfer them to all $m < \omega$. In the case for ω, they paste together the structures for the $\mathbf{0}^{(m)}$. In this paper, we give direct constructions by effective transfinite recursion that extend their results through the hyperarithmetic hierarchy. That is, in Theorem 3.1 we show that for every computable ordinal α, $0^{(\alpha)}$ is a strong degree of categoricity, and in Theorem 3.2 that, given the additional assumption that α is a successor ordinal, so are all degrees $\mathbf{d}$ that are 2-c.e. in and above $0^{(\alpha)}$. We then go on to show in Theorem 4.1 that any degree of categoricity must be hyperarithmetic. We use this to show that the index set of structures with degrees of categoricity, which at first glance seems no better than Σ^1_2, is actually Π^1_1. We then use our result that degrees of the form $\mathbf{0}^{(\alpha)}$ are all degrees of categoricity to show that this index set is actually Π^1_1 complete in Theorem 4.2.

1.1. **Notation and conventions.** As many of the results in this paper use structures defined in Hirschfeldt and White [4], we will adopt many of their conventions. In particular, we will use Ash and Knight's [1] terminology in discussing the hyperarithmetic hierarchy. For general references see Harizanov [3] for computable structure theory, Soare [8] for computability theory and Sacks [7] for hyperarithmetic theory.

Definition 1.3. A system of *notations* for ordinals is comprised of a subset O of the natural numbers, a function $||_O$ that maps each element of O to an ordinal, and a strict partial order $<_O$ on O. In particular,

(1) O contains 1, and $|1|_O = 0$.
(2) If $a \in O$ is a notation for the ordinal α, then $2^a \in O$ and $|2^a|_O = \alpha + 1$. For $b \in O$, we let $b <_O 2^a$ if either $b <_O a$ or $b = a$.
(3) Given a limit ordinal λ, we say that the e^{th} partial computable function φ_e defines a fundamental sequence for λ if it is total, $\varphi_e(k) \leq_O \varphi_e(k+1)$ for all k, and λ is the least upper bound of the ordinals $|\varphi_e(k)|_O$. In this case, $3 \cdot 5^e \in O$ and $|3 \cdot 5^e|_O = \lambda$. For $b \in O$ we will set $b <_O 3 \cdot 5^e$ if there exists k such that $b <_O \varphi_e(k)$.

For technical convenience, Hirschfeldt and White also require in [4] that the fundamental sequences contain only successor ordinals and have 1 as a first element. We assume without loss of generality that all members of O have this property.

Definition 1.4. For $a \in O$ we define $H(a)$ by effective transfinite recursion as follows:
(1) $H(1) = \emptyset$,
(2) $H(2^a) = H(a)'$, and
(3) $H(3{\cdot}5^e) = \{\langle u, v\rangle \mid u <_O 3{\cdot}5^e \ \&\ v \in H(u)\} = \{\langle u, v\rangle \mid \exists n(u \leq_O \varphi_e(n) \ \&\ v \in H(u))\}$.

It is a result of Spector, see [1], that for a computable ordinal α (i.e., one with a notation in O), the Turing degree of $H(a)$ when a is a notation for α does not depend on the choice of a. It is denoted by $\mathbf{0}^{(\alpha)}$.

Now we define the hyperarithmetic hierarchy in terms of computable infinitary formulas.

Definition 1.5. A Σ_0 (Π_0) index for a computable predicate $\mathcal{R}(f, n)$ is a triple $\langle \Sigma, 0, e\rangle$ ($\langle \Pi, 0, e\rangle$) such that e is an index for $\mathcal{R}$. If α is a computable ordinal, a Σ_α index for a predicate $\mathcal{R}(f, n)$ is a triple $\langle \Sigma, a, e\rangle$ such that a is a notation for α and e is an index for a c.e. set

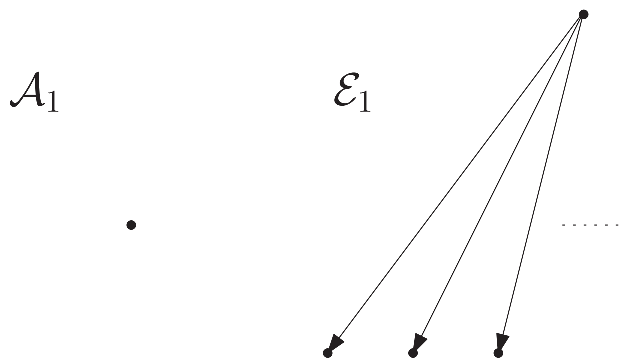

FIGURE 1. $\mathcal{A}_1$ and $\mathcal{E}_1$.

of Π_{β_k} indices for predicates $\mathcal{Q}_k(f,n,x)$ such that $\beta_k < \alpha$ for all $k < \omega$ and

$$\mathcal{R}(f,n) \Leftrightarrow \bigvee_{k<\omega} (\exists x)\mathcal{Q}_k(f,n,x).$$

A Π_α index for a predicate $\mathcal{R}(f,n)$ is a triple $\langle \Pi, a, e\rangle$ such that a is a notation for α and e is an index for a c.e. set of Σ_{β_k} indices for predicates $\mathcal{Q}_k(f,n,x)$ such that $\beta_k < \alpha$ for all $k < \omega$ and

$$\mathcal{R}(f,n) \Leftrightarrow \bigwedge_{k<\omega} (\forall x)\mathcal{Q}_k(f,n,x).$$

We say that a predicate is Σ_α (Π_α) if it has a Σ_α (Π_α) index and Δ_α if it is both Σ_α and Π_α.

2. Hirschfeldt and White's "back-and-forth trees"

The structures in this paper will be built using Hirschfeldt and White's " back-and-forth trees." We use the construction as in [4]; however, instead of fixing a path $O' \subset O$, we define structures $\mathcal{A}_{2^b}$ and $\mathcal{E}_{2^b}$ for each $b \in O$, and other structures at limit ordinals. The structures $\mathcal{A}_a$ and $\mathcal{E}_a$ will be constructed by effective transfinite recursion with a case division as to whether $a = 1$, $a = 2^{2^b}$ (is the successor of a successor), or $a = 2^{3\cdot 5^e}$ (is the successor of a limit). Auxiliary structures $\mathcal{L}^a_\infty$ and $\mathcal{L}^a_k$ for $k < \omega$ will be constructed for $a = 3\cdot 5^e$.

$\mathcal{A}_1$ consists of a single node, and $\mathcal{E}_1$ consists of a single root node with infinitely many children, none of which have children themselves.

Now consider $a = 2^{2^b}$, *i.e.*, the successor of a successor ordinal. $\mathcal{A}_a$ will consist of a single root node with infinitely many copies of $\mathcal{E}_{2^b}$ attached to it, and $\mathcal{E}_a$ will consist of a single root node with infinitely many copies of $\mathcal{A}_{2^b}$ and infinitely many copies of $\mathcal{E}_{2^b}$ attached to it.

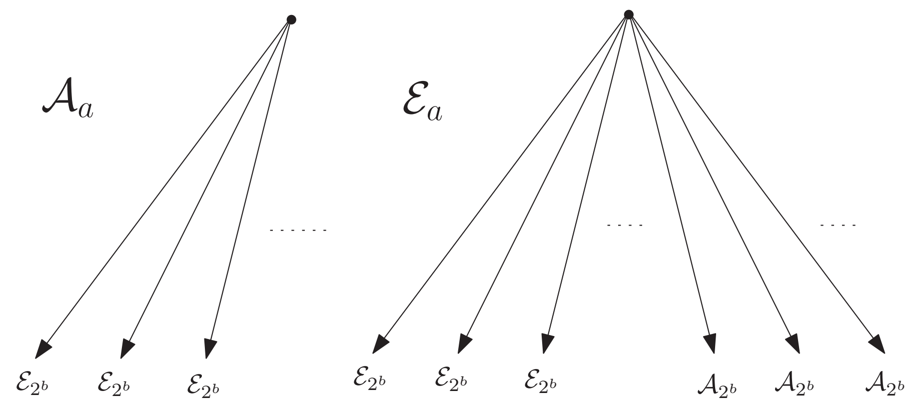

FIGURE 2. $\mathcal{A}_a$ and $\mathcal{E}_a$ for $a = 2^{2^b}$.

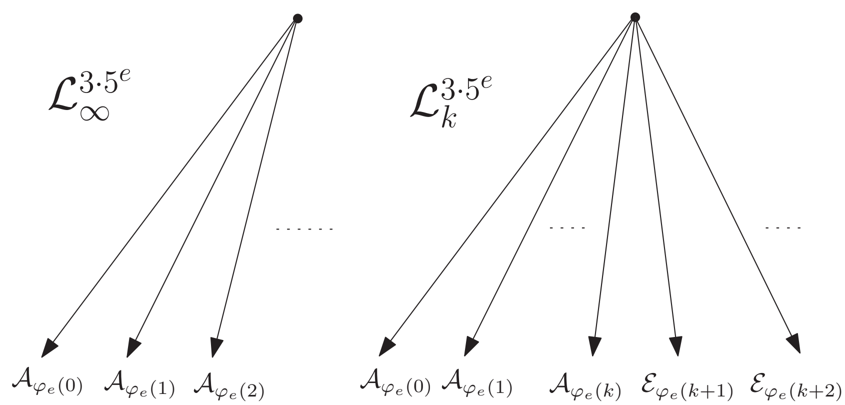

FIGURE 3. $\mathcal{L}_\infty^{3\cdot 5^e}$ and $\mathcal{L}_k^{3\cdot 5^e}$.

Finally, consider an ordinal of the form $a = 2^{3\cdot 5^e}$. To define $\mathcal{A}_a$ and $\mathcal{E}_a$, we first define auxiliary trees $\mathcal{L}_k^{3\cdot 5^e}$ for each $k < \omega$ and an auxiliary tree $\mathcal{L}_\infty^{3\cdot 5^e}$. For each $k < \omega$, we let $\mathcal{L}_k^{3\cdot 5^e}$ consist of a single root node with exactly one copy of $\mathcal{A}_{\varphi_e(n)}$ attached to it for every $n \leq k$ and exactly one copy of $\mathcal{E}_{\varphi_e(n)}$ attached to it for every $n > k$. We define $\mathcal{L}_\infty^{3\cdot 5^e}$ to consist of a single root node with exactly one copy of $\mathcal{A}_{\varphi_e(n)}$ attached to it for every $n < \omega$. (Note: This is where we make use of our assumption that if $3 \cdot 5^e \in O$ then each $\varphi_e(n)$ is a notation for a successor ordinal.)

Now we can define $\mathcal{A}_a$ and $\mathcal{E}_a$. We let $\mathcal{A}_a$ consist of a root node with infinitely many copies of $\mathcal{L}_n^{3\cdot 5^e}$ attached to it for every $n < \omega$ and $\mathcal{E}_a$ consist of a root node with infinitely many copies of $\mathcal{L}_\infty^{3\cdot 5^e}$ and infinitely many copies of $\mathcal{L}_n^{3\cdot 5^e}$ for each $n < \omega$ attached to it.

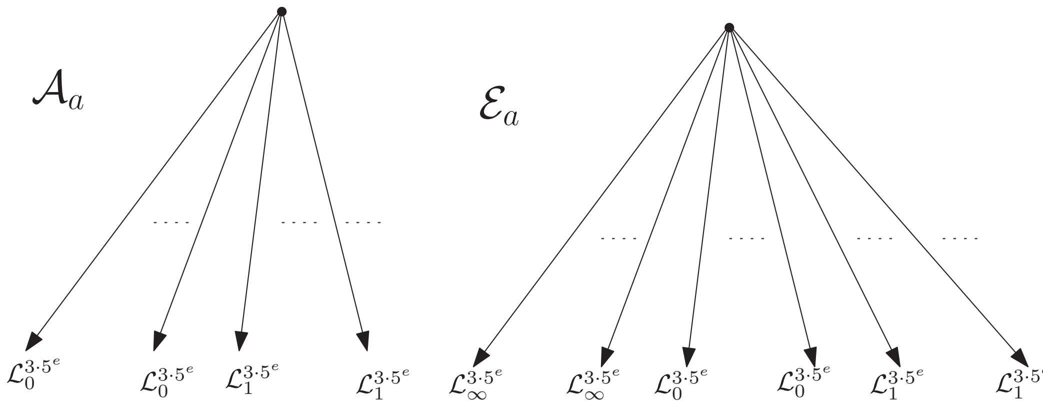

FIGURE 4. $\mathcal{A}_a$ and $\mathcal{E}_a$ for $a = 2^{3\cdot5^e}$.

We say that the back-and-forth trees $\mathcal{L}_\infty^{3\cdot5^e}$ and $\mathcal{L}_n^{3\cdot5^e}$ for $n < \omega$ have rank $3 \cdot 5^e$ and that the back-and-forth trees $\mathcal{A}_a$ and $\mathcal{E}_a$ have rank a.

The above definitions are given by computable transfinite recursion. In fact, we have given a procedure that, for any natural number a, assumes that $a \in O$ with all fundamental sequences containing only successors and begins to build the desired structure. In any case, a computable tree is constructed, and if $a \in O$, then it is of the desired form.

Note that for $a \neq a'$, we may have, for example, $\mathcal{E}_a \not\cong \mathcal{E}_{a'}$ even though $|a|_O = |a'|_O$. Nonetheless, in cases where it is not likely to lead to confusion, we will use $\mathcal{E}_\alpha$, $\mathcal{A}_\alpha$, $\mathcal{L}_k^\alpha$ and $\mathcal{L}_\infty^\alpha$ to denote copies of $\mathcal{E}_a$, $\mathcal{A}_a$, $\mathcal{L}_k^a$ and $\mathcal{L}_\infty^a$, respectively, for some $a \in O$ with $|a|_O = \alpha$.

We will make use of the following results from Hirschfeldt and White [4]. Note that we have slightly reworded their results. Hirschfeldt and White fix a particular path through O, noting that their results work equally well for any path through O, and word their results in terms of computable ordinals α. For us, it will be more convenient to speak of the notations, $a \in O$.

Proposition 2.1 (Proposition 3.2, [4]). *Let $\mathcal{P}(n)$ be a Σ_α predicate.*

(1) *If α is a successor ordinal, then for every notation a for α, there is a sequence of trees $\mathcal{T}_n$, uniformly computable from the notation a and a Σ_α index for $\mathcal{P}$, such that for all n,*

$$\mathcal{T}_n \cong \begin{cases} \mathcal{E}_a & \textit{if } \mathcal{P}(n) \textit{ and} \\ \mathcal{A}_a & \textit{otherwise.} \end{cases}$$

(Note: If $\alpha <_O \omega$, then $\mathcal{P}(n)$ must be $\Sigma_{\alpha+1}$.)

(2) *If α is a limit ordinal, then for every notation a for α, there is a sequence of trees $\mathcal{T}_n$, uniformly computable from the notation a and a Σ_α index for $\mathcal{P}$, such that for all n,*

$$\mathcal{T}_n \cong \begin{cases} \mathcal{L}^a_\infty & \text{if } \neg\mathcal{P}(n) \text{ and} \\ \mathcal{L}^a_k \text{ for some } k & \text{otherwise.} \end{cases}$$

For the following two lemmas, we need to define a *limb* of a tree. We say that a tree $\mathcal{S}$ is a limb of another tree $\mathcal{T}$ if $\mathcal{S} \subseteq \mathcal{T}$ and every child in $\mathcal{T}$ of a node of $\mathcal{S}$ is in $\mathcal{S}$ as well. A *back-and-forth limb* is a limb that is isomorphic to one of Hirschfeldt and White's back-and-forth trees.

Lemma 2.2 (Lemma 3.4, [4])**.** *Let $\mathcal{T}$ be any tree. For each $a \in O$, there is an infinitary formula $\chi_a(x) \in \mathcal{L}_{\omega_1,\omega}$ such that for any back-and-forth limb $\mathcal{S}$ of $\mathcal{T}$ with root r,*

$$\mathcal{T} \models \chi_a(r) \Leftrightarrow \mathit{rank}(\mathcal{S}) = a.$$

Furthermore, "$\mathcal{T} \models \chi_a(r)$" is a Π_α condition for a computable structure $\mathcal{T}$, where $\alpha = |a|_O$, and an index for $\chi_a(x)$ can be found uniformly in a.

If $\mathcal{S}$ is a back-and-forth limb, then by definition it is isomorphic to $\mathcal{A}_a$, $\mathcal{E}_a$, $\mathcal{L}^a_\infty$ or $\mathcal{L}^a_k$ for some $a \in O$. We call the isomorphism type of $\mathcal{S}$ a *back-and-forth index of* $\mathcal{S}$ and note that it can be viewed as a natural number (namely, the computable index of the back-and-forth tree it is isomorphic to). Note that a back-and-forth limb may have more than one rank, and more than one back-and-forth index, as there might be distinct yet isomorphic back-and-forth trees. However, along a fixed path of O, the ranks and back-and-forth indices are unique, and in such cases we may speak of "*the* rank" and "*the* back-and-forth index".

Definition 2.3. To each back-and-forth tree $\mathcal{B}$, we associate a *natural complexity* based on its back-and-forth index as follows. For $n < \omega$, $\mathcal{E}_n$ has natural complexity Σ_{n+1} and $\mathcal{A}_n$ has natural complexity Π_{n+1}. For $\alpha \geq \omega$, $\mathcal{E}_\alpha$ has natural complexity Σ_α, $\mathcal{A}_\alpha$ has natural complexity Π_α, $\mathcal{L}^\alpha_\infty$ has natural complexity Π_α, and $\mathcal{L}^\alpha_k$ has natural complexity Σ_α.

Lemma 2.4 (Lemma 3.5, [4])**.** *Let $\mathcal{T}$ be a tree, and let $\mathcal{B}$ be any back-and-forth tree. Then there is an infinitary formula $\varphi_{\mathcal{B}}(x) \in \mathcal{L}_{\omega_1,\omega}$ such that for any back-and-forth limb $\mathcal{S}$ of $\mathcal{T}$ which has root r and the same rank as $\mathcal{B}$,*

$$\mathcal{T} \models \begin{cases} \varphi_{\mathcal{B}}(r) & \text{if } \mathcal{S} \cong \mathcal{B} \text{ and} \\ \neg\varphi_{\mathcal{B}}(r) & \text{otherwise.} \end{cases}$$

Furthermore, for computable $\mathcal{T}$, the complexity of "$\mathcal{T} \models \varphi_{\mathcal{B}}(r)$" is the natural complexity of $\mathcal{B}$, and an index for $\varphi_{\mathcal{B}}(r)$ can be found uniformly from the back-and-forth index of $\mathcal{B}$.

We can use the above lemmas to see to what extent $\mathbf{0}^{(\alpha)}$ can be used to compute isomorphisms between back-and-forth trees.

Corollary 2.5. *Let $\mathcal{T}$ be any computable tree, and let $\mathcal{S}$ be a back-and-forth limb of $\mathcal{T}$ with rank($\mathcal{S}$) $<_O a$, and root r. Then $H(a)$ can, uniformly in r, compute the back-and-forth index of $\mathcal{S}$.*

Proof. Recall that $\{b \mid b <_O a\}$ is c.e. By Lemma 2.2, for each $b \in O$, $\mathcal{T} \models \chi_b(r)$ is a $\Pi_{|b|_O}$ condition, so for $b <_O a$, $H(a)$ can compute whether $\mathcal{T} \models \chi_b(r)$. As $\mathcal{T} \models \chi_b(r)$ for some $b <_O a$, $H(a)$ can compute $b =$rank$(\mathcal{S})$. Now, for each back-and-forth tree $\mathcal{B}$ of rank b, by Lemma 2.4, $\mathcal{T} \models \varphi_{\mathcal{B}}(r)$ has complexity Π_b or Σ_b. In either case, since $b <_O a$, $H(a)$ can compute whether $\mathcal{T} \models \varphi_{\mathcal{B}}(r)$ and thus compute the back-and-forth index of $\mathcal{S}$. $\square$

Corollary 2.6. *Let $\mathcal{S}_1$ be a back-and-forth limb of a computable tree $\mathcal{T}_1$ with root r_1, and let $\mathcal{S}_2$ be a back-and-forth limb of a computable tree $\mathcal{T}_2$ with root r_2. Suppose that $\mathcal{S}_1 \cong \mathcal{S}_2$ and rank($\mathcal{S}_i$)$\leq_O a$. Then $H(a)$ can, uniformly in r_1 and r_2, compute an isomorphism $f : \mathcal{S}_1 \to \mathcal{S}_2$.*

Proof. This follows easily by Corollary 2.5 and recursive transfinite induction. Indeed, suppose the result holds for all $b <_O a$. Let $c^i_1, c^i_2, ...$ denote the children of r_i in $\mathcal{S}_i$. Note that since $H(a) \geq_T \emptyset'$, certainly $H(a)$ can uniformly compute the c^i_j. Since rank$(\mathcal{S}_i) \leq_O a$, each c^i_j is the root of a back-and-forth limb of rank $<_O a$. By Corollary 2.5, $H(a)$ can uniformly compute the back-and-forth index of the limb with root c^i_j, so $H(a)$ can bijectively match each c^1_j to some c^2_k with the same index (since $\mathcal{S}_1 \cong \mathcal{S}_2$). Then by the induction hypothesis, $H(a)$ can extend this bijection to an isomorphism between $\mathcal{S}_1$ and $\mathcal{S}_2$. $\square$

3. Examples of degrees of categoricity

We begin by demonstrating that certain degrees in the hyperarithmetic hierarchy are degrees of categoricity.

Theorem 3.1. *For any $a \in O$, there is a computable structure $\mathcal{S}_a$ with (strong) degree of categoricity $H(a)$.*

Proof. We build the structures $\mathcal{S}_a$ for $a \in O$ by transfinite recursion. Note that in fact for any $a \in \omega$ the procedure can be used to build a computable structure $\mathcal{S}_a$, where the structure has the desired form for each $a \in O$.

For each $a \in O$, we will build a "standard" copy of $\mathcal{S}_a$ as well as a "hard" but still computable copy $\widehat{\mathcal{S}}_a$ such that any isomorphism between $\mathcal{S}_a$ and $\widehat{\mathcal{S}}_a$ will compute $H(a)$.

First, we consider the case $a = 2$. Let $\mathcal{S}_2$ consist of infinitely many disjoint copies of $\mathcal{A}_1$ and $\mathcal{E}_1$, and fix an approximation $\{K_s\}_{s\in\omega}$ to $0'$. We define the set of edges of the standard copy $\mathcal{S}_2$ to be

$$\{(\langle 2n, 0\rangle, \langle 2n, k\rangle) \mid k > 0\},$$

so the substructure consisting of the elements in an "even" column is isomorphic to $\mathcal{E}_1$. Now we define the set of edges of the hard copy, $\widehat{\mathcal{S}}_2$, to be

$$\{(\langle 2n, 0\rangle, \langle 2n, t\rangle) \mid n \in K_t\}.$$

In $\widehat{\mathcal{S}}_2$, the root nodes of copies of $\mathcal{E}_1$ are of the form $\langle 2n, 0\rangle$ for $n \in K$. Let $f : \widehat{\mathcal{S}}_2 \cong \mathcal{S}_2$. Then $n \in K \iff f(\langle 2n, 0\rangle) = \langle 2m, 0\rangle$ for some m.

Now we consider two arbitrary computable copies of $\mathcal{S}_2$. Since $\mathbf{0}'$ can answer all Σ_1 and Π_1 questions, it can determine which elements of each are $\mathcal{A}_1$ components, which are roots in $\mathcal{E}_1$ components, and which are children in $\mathcal{E}_1$ components, so it is powerful enough to compute an isomorphism between these copies.

Now we consider the case of 2^a where $a = 2^b$ for some $b \in O$. In this case, let $\mathcal{S}_{2^a}$ consist of infinitely many disjoint copies of $\mathcal{A}_a$ and $\mathcal{E}_a$.

We now verify that $\mathcal{S}_{2^a}$ has degree of categoricity $H(2^a)$. In the following discussion, we assume that $|a|_O > \omega$; the case where $|a|_O$ is finite is similar but some indices are off by one.

Since $H(2^a)$ is $\Sigma^0_{|a|_O}$, using Proposition 2.1 (1), we can build a computable copy $\widehat{\mathcal{S}}_{2^a}$ of $\mathcal{S}_{2^a}$ such that for every n, $\langle n, 0\rangle$ is a parent node of a tree isomorphic to $\mathcal{E}_a$ if $n \in H(2^a)$ and a parent node of a tree isomorphic to $\mathcal{A}_a$ if $n \notin H(2^a)$. We have a standard copy of $\mathcal{S}_{2^a}$ where $\langle 2n, 0\rangle$ is a parent node of a tree isomorphic to $\mathcal{E}_a$ and $\langle 2n+1, 0\rangle$ is a parent node of a tree isomorphic to $\mathcal{A}_a$ for every n. Let $f : \widehat{\mathcal{S}}_{2^a} \cong \mathcal{S}_{2^a}$. Then $n \in H(2^a)$ if and only if $f(\langle n, 0\rangle) = \langle 2k, 0\rangle$ for some k.

Conversely, let $\mathcal{B}$ be an arbitrary computable copy of $\mathcal{S}_{2^a}$. We describe an $H(2^a)$ computable isomorphism $f : \mathcal{B} \to \mathcal{S}_{2^a}$. It is a Σ_1 question whether a vertex in $\mathcal{B}$ has an edge going to it. Hence, $H(2^a)$ can certainly compute the root nodes, r_i, of all the connected components of $\mathcal{B}$. Each connected component is isomorphic to either $\mathcal{E}_a$ or $\mathcal{A}_a$, which both have rank a. By Corollary 2.5, $H(2^a)$ uniformly computes the back-and-forth index of the limb extending from each root node. Thus $H(2^a)$ can first define a bijection between the root nodes r_i in $\mathcal{B}$ and the root nodes $\langle i, 0\rangle$ in $\mathcal{S}_{2^a}$ that is back-and-forth index preserving. Then, by Corollary 2.6, $H(2^a)$ can extend this to an isomorphism $\mathcal{B} \cong \mathcal{S}_{2^a}$.

Now we consider the case where $a = 3\cdot 5^e$. Let $\mathcal{S}_a$ consist of infinitely many disjoint copies of $\mathcal{A}_{\varphi_e(k)}$ and $\mathcal{E}_{\varphi_e(k)}$ for all $k \in \omega$.

To make the hard copy, $\widehat{\mathcal{S}}_a$, we proceed as follows. Using Proposition 2.1 (1), let $\langle u, v, n+1, 0\rangle$ be the parent node of a tree that is isomorphic to $\mathcal{E}_{\varphi_e(n)}$ if $u \leq_O \varphi_e(n)$ and $v \in H(u)$ and isomorphic to $\mathcal{A}_{\varphi_e(n)}$ otherwise. Let $\langle u, v, 0, 0\rangle$ be the parent node of a tree that is isomorphic to $\mathcal{E}_{\varphi_e(0)}$ if $(\exists n)[u \leq_O \varphi_e(n)]$ and isomorphic to $\mathcal{A}_{\varphi_e(0)}$ otherwise. To make the standard copy, we let $\langle k, n, 0\rangle$ be the parent node of a tree that is isomorphic to $\mathcal{E}_{\varphi_e(n)}$ if k is even and $\mathcal{A}_{\varphi_e(n)}$ if k is odd.

Let $f : \widehat{\mathcal{S}}_a \cong \mathcal{S}_a$; we wish to use f to compute $H(a)$. First, compute $f(\langle u, v, 0, 0\rangle)$. If $f(\langle u, v, 0, 0\rangle)$ is not of the form $\langle 2y, 0, 0\rangle$, then $u \not<_O a$, so $\langle u, v\rangle \notin H(a)$. Otherwise, we know that $u <_O a$, so we search for n such that $u \leq_O \varphi_e(n)$. Then we compute $f(\langle u, v, n+1, 0\rangle)$, which must have the form $\langle k, n, 0\rangle$ for some k. We have $\langle u, v\rangle \in H(a)$ if and only if k is even.

Now we let $\mathcal{B}$ be an arbitrary computable copy of $\mathcal{S}_a$. We describe an $H(a)$ computable isomorphism $f : \mathcal{B} \to \mathcal{S}_a$. As before, $H(a)$ can certainly compute the root nodes, r_i, of all the connected components of $\mathcal{B}$. Each connected component is a back-and-forth tree of rank $<_O a$. By Corollary 2.5, $H(a)$ uniformly computes the back-and-forth index of the limb extending from each root node. Thus, as before, $H(a)$ can first define a bijection between the root nodes r_i in $\mathcal{B}$ and the root nodes $\langle k, n, 0\rangle$ in $\mathcal{S}_a$ that is back-and-forth index preserving. Then, by Corollary 2.6, $H(a)$ can extend this to an isomorphism $\mathcal{B} \cong \mathcal{S}_a$.

Finally, we consider the case of 2^a where $a = 3\cdot 5^e$. Let $\mathcal{S}_{2^a}$ consist of infinitely many disjoint copies of $\mathcal{L}^a_\infty$ and $\mathcal{L}^a_k$ for all $k < \omega$. To make the standard copy, we let $\langle n, k, 0\rangle$ be the parent node of a tree that is isomorphic to $\mathcal{L}^a_k$ if n is even and $\mathcal{L}^a_\infty$ otherwise. $H(2^a)$ is $\Sigma^0_{|a|_O}$. Using Proposition 2.1 (2), we can build a computable copy $\widehat{\mathcal{S}}_{2^a}$ of $\mathcal{S}_{2^a}$ such that for every n, $\langle n, 0\rangle$ is a parent node of a tree isomorphic to $\mathcal{L}^a_k$ for some k if $n \in H(2^a)$ and a parent node of a tree isomorphic to $\mathcal{L}^a_\infty$ if $n \notin H(2^a)$. Let $f : \widehat{\mathcal{S}}_{2^a} \cong \mathcal{S}_{2^a}$. Then $n \in H(2^a)$ if and only if $f(\langle n, 0\rangle) = \langle 2l, k, 0\rangle$ for some l and k.

The argument that $H(2^a)$ suffices to compute an isomorphism between arbitrary computable presentations of $\mathcal{S}_{2^a}$ is as in the previous cases, since all connected components are back-and-forth trees with rank $<_O 2^a$. $\square$

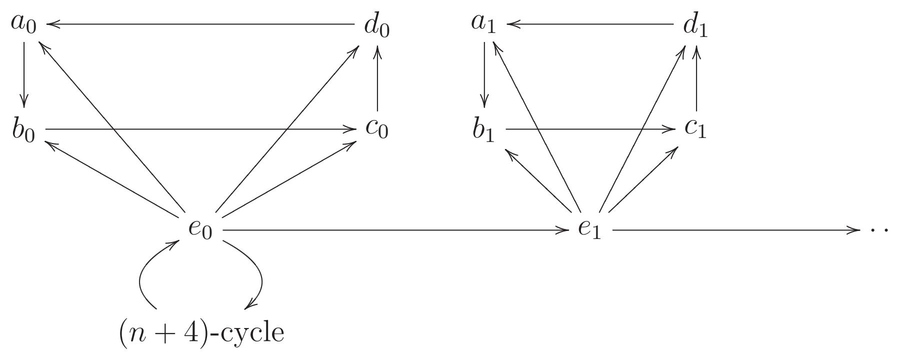

FIGURE 5. Basic structure of $\mathcal{G}$ and $\widehat{\mathcal{G}}$

Now we can extend the set of known degrees of categoricity even further.

Theorem 3.2. *Let α be a computable successor ordinal. If $\mathbf{d}$ is 2-c.e. in and above $\mathbf{0}^{(\alpha)}$, then $\mathbf{d}$ is a (strong) degree of categoricity.*

Proof. We build a graph $\mathcal{G}$ with degree of categoricity $\mathbf{d}$. Let $D \in \mathbf{d}$ be 2-c.e. in and above $\mathbf{0}^{(\alpha)}$, and let B and C be Σ^0_α sets such that $C \subset B$ and $D = B - C$. Let $\{B_k\}_{k\in\omega}$ be an enumeration of B relative to $\mathbf{0}^{(\alpha)}$. For each $n \in \omega$, we will make use of vertices labeled $e^n_k, a^n_k, b^n_k, c^n_k, d^n_k$ for $k \in \omega$, which will all belong to a single connected component of the graph. We attach an $(n+4)$-cycle to the point e^n_0. We now use Proposition 2.1 (1) to build two computable copies $\mathcal{G}$ and $\widehat{\mathcal{G}}$ of the graph. Since the description of the component for n is the same for each n, we drop the superscript to ease notation. In both copies, we have edges (e_k, e_{k+1}), (e_k, a_k), (e_k, b_k), (e_k, c_k), (e_k, d_k), (a_k, b_k), (b_k, c_k), (c_k, d_k), and (d_k, a_k) for all k (Figure 5). In both copies, a_0 will be a parent node of a tree isomorphic to $\mathcal{E}_\alpha$ and c_0 will be a parent node of a tree isomorphic to $\mathcal{E}_\alpha$ if $n \in C$ and $\mathcal{A}_\alpha$ otherwise. In $\mathcal{G}$, let b_0 be a parent node of a tree isomorphic to $\mathcal{E}_\alpha$ if $n \in B$ and $\mathcal{A}_\alpha$ otherwise, and let d_0 be a parent node of a tree isomorphic to $\mathcal{E}_\alpha$ if $n \in C$ and $\mathcal{A}_\alpha$ otherwise (Figure 6). In $\widehat{\mathcal{G}}$, we reverse the roles of b_0 and d_0 (Figure 7).

Now for $k > 0$, in both copies, we will make a_k a parent node of a tree isomorphic to $\mathcal{E}_\alpha$ and c_k a parent node of a tree isomorphic to $\mathcal{A}_\alpha$. In $\mathcal{G}$, let b_k be a parent node of a tree isomorphic to $\mathcal{E}_\alpha$ if $n \in (B_{k+1} - B_k) \cap C$ and $\mathcal{A}_\alpha$ otherwise, and let d_k be a parent node of a tree isomorphic to $\mathcal{A}_\alpha$ (Figure 8). In $\widehat{\mathcal{G}}$, we reverse the roles of b_k and d_k (Figure 9).

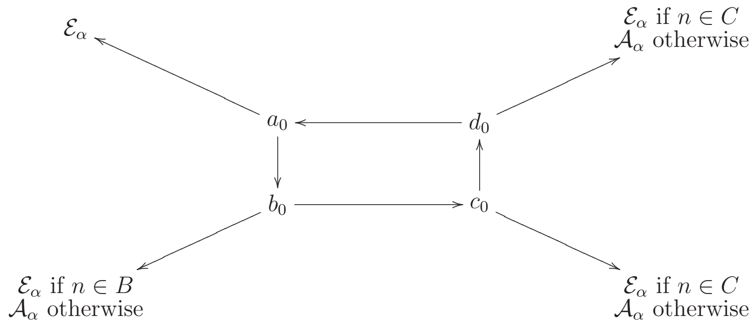

FIGURE 6. $\mathcal{G}$

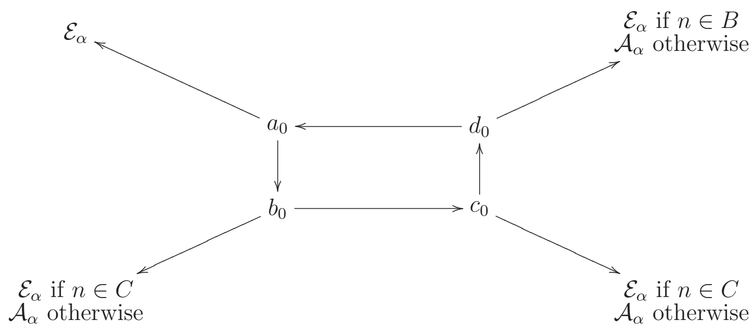

FIGURE 7. $\widehat{\mathcal{G}}$

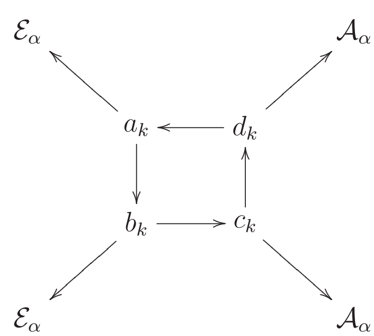

FIGURE 8. $\mathcal{G}$ for $k > 0$ if $n \in (B_{k+1} - B_k) \cap C$

Finally, put a 3-cycle where each point has an edge to each point of a copy of $\mathcal{S}_\alpha$ (built the "standard" way) into $\mathcal{G}$, and a 3-cycle attached to a copy of $\widehat{\mathcal{S}}_\alpha$ (built the "hard" way) into $\widehat{\mathcal{G}}$.

$$\begin{tikzcd}
\mathcal{E}_\alpha & & & \mathcal{E}_\alpha \\
& a_k \arrow[ul] \arrow[d] & d_k \arrow[l] \arrow[ur] & \\
& b_k \arrow[r] \arrow[dl] & c_k \arrow[u] \arrow[dr] & \\
\mathcal{A}_\alpha & & & \mathcal{A}_\alpha
\end{tikzcd}$$

FIGURE 9. $\widehat{\mathcal{G}}$ for $k > 0$ if $n \in (B_{k+1} - B_k) \cap C$

Note that in any computable copy of the structure, for each n, the points isomorphic to $e_0, e_1, e_2, \ldots$ are computable, as are the 4-cycles emanating from them. Each member of such a 4-cycle is a parent node of a tree isomorphic to either $\mathcal{E}_\alpha$ or $\mathcal{A}_\alpha$, and an isomorphism of two copies of the structure matches these up correctly.

Let p be an isomorphism between $\mathcal{G}$ and $\widehat{\mathcal{G}}$. Now if $p(a_0) = a_0$, then either $n \in C$, in which case $n \notin D$, or $n \notin B$, in which case $n \notin D$. Therefore, if $p(a_0) = a_0$, then $n \notin D$. If $p(a_0) \neq a_0$, then $n \in B$. This means that $n \in D$ if and only if $n \notin C$. Since p computes $H(\alpha)$ (since it computes an isomorphism between $\mathcal{S}_\alpha$ and $\widehat{\mathcal{S}}_\alpha$) and since $n \in B$, p computes the k such that $n \in B_{k+1} - B_k$. Then $n \in D$ if and only if $p(a_k) = a_k$.

Conversely, we claim that, given arbitrary computable copies of the structures, D can compute an isomorphism between them. Note that the limbs of $\mathcal{E}_\alpha$ and $\mathcal{A}_\alpha$ all have rank $\alpha - 1$, so by Corollary 2.5, $H(\alpha)$ can compute the back-and-forth index of limbs of the trees attached to any of a_k, b_k, c_k, d_k in either copy. Moreover, $\mathcal{E}_\alpha$ has $\mathcal{A}_\alpha$ as a proper substructure; i.e., the any back-and-forth index of a limb of $\mathcal{A}_\alpha$ also occurs as the back-and-forth index of a limb of $\mathcal{E}_\alpha$, but not conversely. Thus the fact that a tree attached to some node is $\mathcal{E}_\alpha$ (and not $\mathcal{A}_\alpha$) is c.e. in $H(\alpha)$.

If $n \notin D$, then there are two possibilities: either n is not in B (and therefore, not in C) or n is in both B and C. In the first case, exactly one of the nodes a_0, b_0, c_0, and d_0 (in fact, a_0) is a parent node of a tree isomorphic to $\mathcal{E}_\alpha$ (Figure 10). In the second case, all of these nodes are parent nodes of a tree isomorphic to $\mathcal{E}_\alpha$ (Figure 11). Now, by Corollary 2.5, we can use $H(\alpha)$ to find at least one $\mathcal{E}_\alpha$ parent node in each copy and match these up, applying Corollary 2.5 to match up limbs of the $\mathcal{E}_\alpha$

$$\begin{tikzcd}
\mathcal{E}_\alpha & & & \mathcal{A}_\alpha \\
& a_0 \arrow[ul] \arrow[d] & d_0 \arrow[l] \arrow[ur] & \\
& b_0 \arrow[r] \arrow[dl] & c_0 \arrow[u] \arrow[dr] & \\
\mathcal{A}_\alpha & & & \mathcal{A}_\alpha
\end{tikzcd}$$

FIGURE 10. $\mathcal{G}$ and $\widehat{\mathcal{G}}$ if $n \notin B$

$$\begin{tikzcd}
\mathcal{E}_\alpha & & & \mathcal{E}_\alpha \\
& a_0 \arrow[ul] \arrow[d] & d_0 \arrow[l] \arrow[ur] & \\
& b_0 \arrow[r] \arrow[dl] & c_0 \arrow[u] \arrow[dr] & \\
\mathcal{E}_\alpha & & & \mathcal{E}_\alpha
\end{tikzcd}$$

FIGURE 11. Copy 1 and Copy 2 if $n \in C$

with the same back-and-forth index, and Corollary 2.6 to (uniformly) extend these to isomorphisms of the limbs. The rest of the four nodes can then be safely matched up, using the same procedure to construct the isomorphism, assuming they are all copies of $\mathcal{A}_\alpha$—if at some point it turns out that one is an $\mathcal{E}_\alpha$ and not an $\mathcal{A}_\alpha$, then they all are. If $n \in D$, then exactly two of the nodes a_0, b_0, c_0 and d_0 are parent nodes of a tree isomorphic to $\mathcal{E}_\alpha$ (Figures 12 and 13); $H(\alpha)$ can find these in each copy and match them up.

Finally, since $D \geq_T H(\alpha)$, D can compute an isomorphism between the copies of $\mathcal{S}_\alpha$ attached to the 3-cycle in each copy.

For $k > 0$, exactly one or two of the nodes a_k, b_k, c_k, d_k are parent nodes of a tree isomorphic to $\mathcal{E}_\alpha$. The only way that there are two is if $n \in (B_{k+1} - B_k) \cap C$. Now $H(\alpha)$ and hence D can compute whether $n \in B_{k+1} - B_k$, and if $n \in B_{k+1} - B_k$, then $n \in C$ if and only if $n \notin D$. Therefore, D can compute whether there are one or two copies of $\mathcal{E}_\alpha$

$$
\begin{tikzcd}
\mathcal{E}_\alpha & & & \mathcal{A}_\alpha \\
& a_0 \arrow[ul] \arrow[d] & d_0 \arrow[l] \arrow[ur] & \\
& b_0 \arrow[r] \arrow[dl] & c_0 \arrow[u] \arrow[dr] & \\
\mathcal{E}_\alpha & & & \mathcal{A}_\alpha
\end{tikzcd}
$$

FIGURE 12. Copy 1 if $n \in B$ and $n \notin C$

$$
\begin{tikzcd}
\mathcal{E}_\alpha & & & \mathcal{E}_\alpha \\
& a_0 \arrow[ul] \arrow[d] & d_0 \arrow[l] \arrow[ur] & \\
& b_0 \arrow[r] \arrow[dl] & c_0 \arrow[u] \arrow[dr] & \\
\mathcal{A}_\alpha & & & \mathcal{A}_\alpha
\end{tikzcd}
$$

FIGURE 13. Copy 2 if $n \in B$ and $n \notin C$

with parent nodes among a_k, b_k, c_k, d_k, and $H(\alpha)$ can find these in each copy and match them up. □

4. General properties of degrees of categoricity

Fokina, Kalimullin, and R. Miller showed that any strong degree of categoricity is hyperarithmetical. The Effective Perfect Set Theorem is the main ingredient in their proof [2]. Here, we strengthen their result and show that every degree of categoricity is hyperarithmetic. We then go on to show that the index set of structures that have a degree of categoricity is actually Π^1_1 complete.

Theorem 4.1. *If* $\mathbf{d}$ *is a degree of categoricity, then* $\mathbf{d} \in HYP$.

Proof. Let $\mathbf{d} \notin HYP$, and let $\mathcal{A}$ be any computable structure. We will show that $\mathbf{d}$ is not a degree of categoricity for $\mathcal{A}$. Let $\mathcal{A}_0$, $\mathcal{A}_1$, $\mathcal{A}_2$,... be a list of all computable copies of $\mathcal{A}$. Note that $\{f \mid f : \mathcal{A}_0 \cong \mathcal{A}_1\}$ is Π^0_2, so it is certainly Σ^1_1. Hence by Kreisel's Basis Theorem [7], there exists an isomorphism $f_1 : \mathcal{A}_0 \cong \mathcal{A}_1$ such that $\mathbf{d} \not\leq_h f_1$. Suppose we are given isomorphisms $f_i : \mathcal{A}_0 \cong \mathcal{A}_i$ for $1 \leq i \leq n$ such that $\mathbf{d} \not\leq_h f_1 \oplus \cdots \oplus f_n$. Then by Kreisel's Basis Theorem relativized to $f_1 \oplus \cdots \oplus f_n$, there exists an isomorphism $f_{n+1} : \mathcal{A}_0 \cong \mathcal{A}_{n+1}$ such that $\mathbf{d} \not\leq_h f_1 \oplus \cdots \oplus f_{n+1}$. Now let $\mathbf{a}, \mathbf{b}$ be an exact pair for the sequence $f_1, f_1 \oplus f_2, f_1 \oplus f_2 \oplus f_3, ...$ Since f_n is an isomorphism between $\mathcal{A}_0$ and $\mathcal{A}_n$, both $\mathbf{a}$ and $\mathbf{b}$ can compute an isomorphism between any two arbitrary computable copies of $\mathcal{A}$. Therefore, if $\mathbf{d}$ is a degree of categoricity for $\mathcal{A}$, then $\mathbf{d} \leq \mathbf{a}$ and $\mathbf{d} \leq \mathbf{b}$. However, since $\mathbf{a}, \mathbf{b}$ is an exact pair, $\mathbf{d} \leq_T f_1 \oplus \cdots \oplus f_n$ for some n, giving us a contradiction. □

Now we consider the complexity of the index set of structures that have a degree of categoricity. Let $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ be a list of the partial computable structures.

Theorem 4.2. *The index set* $\mathcal{DC} = \{e \mid \mathcal{A}_e$ *has a degree of categoricity*$\}$ *is* Π^1_1 *complete.*

Proof. We begin by giving the natural formula $\psi(e)$ that expresses that the structure $\mathcal{A}_e$ has a degree of categoricity, that is, the formula

$$(\exists D)\{(\forall i)[(\exists F)(F : \mathcal{A}_e \cong \mathcal{A}_i) \quad \rightarrow \quad (\exists \hat{F} \leq_T D)(\hat{F} : \mathcal{A}_e \cong \mathcal{A}_i)] \ \& $$
$$(\forall C)[(\forall i)[(\exists F)(F : \mathcal{A}_e \cong \mathcal{A}_i) \quad \rightarrow \quad (\exists \hat{F} \leq_T C)(\hat{F} : \mathcal{A}_e \cong \mathcal{A}_i) \rightarrow D \leq_T C]\}.$$

We note that $\psi(e)$ is Σ^1_3. However, by Theorem 4.1, we know that any degree of categoricity is hyperarithmetic. Thus many of the existential quantifiers in ψ can be bounded by HYP without changing the meaning of the formula. That is, we have the following formula $\hat{\psi}(e)$ expressing that $\mathcal{A}_e$ has a degree of categoricity.

$$(\exists D \in HYP)\{(\forall i)[(\exists F)(F : \mathcal{A}_e \cong \mathcal{A}_i) \quad \rightarrow \quad (\exists \hat{F} \leq_T D)(\hat{F} : \mathcal{A}_e \cong \mathcal{A}_i)] \ \& $$
$$(\forall C)[(\forall i)[(\exists F \in HYP)(F : \mathcal{A}_e \cong \mathcal{A}_i) \quad \rightarrow \quad (\exists \hat{F} \leq_T C)(\hat{F} : \mathcal{A}_e \cong \mathcal{A}_i) \rightarrow D \leq_T C]\}$$

We observe that this formula is Π^1_1: quantifiers like $\exists D \in HYP$ can be written as Π^1_1 formulas, and quantifiers like $\exists \hat{F} \leq_T D$ are arithmetic. Furthermore, we note that we can write $\exists F \in HYP$ in the second half of the conjunction instead of simply $\exists F$ because we require D to be in HYP and because the first half of the conjunction states that there is an isomorphism from $\mathcal{A}_e$ to $\mathcal{A}_i$ that is computable from D. Therefore, $\hat{\psi}(e)$ is equivalent to $\psi(e)$. Now we only need to show that the set $\mathcal{DC}$ is Π^1_1 complete.

For each $a \in \omega$, we will, uniformly in a, define a computable structure $\mathcal{R}_a$. If $a \in O$, then $\mathcal{R}_a$ will have a degree of categoricity. If $a \notin O$ then

$\mathcal{R}_a$ will have distinguishable computable substructures with degrees of categoricity $H(b)$ for b's in O of rank unbounded in ω_1^{CK}. Thus $\mathcal{R}_a$ itself not have a degree of categoricity by Theorem 4.1.

We begin with a folklore reduction that can be found in [9], Prop. 5.4.1: There is a recursive function f such that if $a \in O$ then $f(a)$ is an index for a recursive linear ordering of type $\alpha < \omega_1^{CK}$ and if $a \notin O$ then $f(a)$ is an index for a recursive linear ordering of type $\omega_1^{CK}(1+\eta)$ (the Harrison linear ordering). We then apply a standard translation g of a recursive linear order into a recursive notation system $\leq_{g(e)}$ as can be found in [6] (Theorem 11.8.XX). This translation takes an index e for a recursive well ordering to a recursive notation system in O of limit ordinal length (actually ω times the length of the original ordering) and one for a linear order of type $\omega_1^{CK}(1+\eta)$ to a recursive notation system (although not a well founded one) with a well founded initial segment that is a path in O of type ω_1^{CK}.

Let $\mathcal{R}_a$ be a disjoint labeled union of the $\mathcal{S}_b$ for each b in the notation system $g(f(a))$, i.e. there is a $(b+3)$-cycle with a copy of $\mathcal{S}_b$ attached to a node in the cycle. Note that since the notation system is recursive and each $\mathcal{S}_b$ is uniformly computable (in b), $\mathcal{R}_a$ is computable. Note that we can build "standard" and "hard" copies of $\mathcal{R}_a$ by using (for any $b \in O$) the standard copies of $\mathcal{S}_b$ in the standard copy of $\mathcal{R}_a$ and the hard copies $\widehat{\mathcal{S}}_b$ in the hard copy $\widehat{\mathcal{R}}_a$ of $\mathcal{R}_a$. Since for each b in the system, the copy of $\mathcal{S}_b$ in $\mathcal{R}_a$ is labeled by a $(b+3)$-cycle, it is easy to see that for any $b \in O$ in the system, any isomorphism $f : \widehat{\mathcal{R}}_a \to \mathcal{R}_a$ computes $H(b)$. If $a \in O$, then the notation system is an initial segment of O of length some limit notation $c \in O$. By the uniformities of all our constructions, it is easy to see that $H(c)$ can always compute an isomorphism between any two copies of $\mathcal{R}_a$. On the other hand, the uniform replacement of the $\mathcal{S}_b$ by hard and easy ones produces copies such that any isomorphism uniformly computes $H(b)$ for every $b <_O c$, and so computes $H(c)$. Hence if $a \in O$, $\mathcal{R}_a$ has degree of categoricity $H(c)$. On the other hand, if $a \notin O$ then the system includes initial segments of a path in O of length ω_1^{CK}. Thus $\mathcal{R}_a$ includes distinguishable copies of $\mathcal{S}_b$ for b of rank unbounded in ω_1^{CK}. Again we can build "easy" and "hard" copies of $\mathcal{R}_a$ by using "easy" and "hard" copies of the $\mathcal{S}_b$ as appropriate. We then see that any degree of categoricty of $\mathcal{R}_a$ must compute $H(b)$ for each b in the system $g(f(a))$. As these are of unbounded rank, it follows that the degree of categoricity cannot be hyperarithmetical, contradicting Theorem 4.1.

□

5. Further questions

Although we have answered some questions arising from Fokina, Kalimullin, and Miller's work, several questions still remain.

Question 5.1. *Is every degree that is n-c.e. in and above a degree of the form $\mathbf{0}^{(\alpha)}$ for a computable ordinal α and some $n < \omega$ a (strong) degree of categoricity?*

Fokina, Kalimullin and Miller answered the above, in the affirmative and for strong degrees, for all $\alpha \leq \omega$ and $n \leq 2$. In this paper, we have extended the result for $n \leq 2$ and all computable successor ordinals α. For $n = 2$ the question is open for limit ordinals, and for $n > 2$ it is open for any computable α.

So far, all known degrees of categoricity are 2-c.e. in and above $\mathbf{0}^{(\alpha)}$ for a computable ordinal α.

Question 5.2. *Is there a degree of categoricity that is not n-c.e. in and above a degree of the form $\mathbf{0}^{(\alpha)}$ for a computable ordinal α and some $n < \omega$?*

Finally, the question of whether the degrees of categoricity are precisely the strong degrees of categoricity are the same is still open.

Question 5.3. *Is every degree of categoricity a strong degree of categoricity?*

Theorem 4.1 tells us that we can limit our search for a counterexample to the hyperarithmetic case and provides some evidence that these degrees are the same, but no more.

Along similar lines, but with a view towards the structures, one might ask the following.

Question 5.4. *Does there exist a structure which has a degree of categoricity that is not a strong degree of categoricity for that structure?*

References

[1] C.J. Ash and J. Knight. *Computable Structures and the Hyperarithmetical Hierarchy*. Number 144 in Studies in Logic and the Foundations of Mathematics. North-Holland, 2000.

[2] Ekaterina B. Fokina, Iskander Kalimullin, and Russell Miller. Degrees of categoricity of computable structures. *Arch. Math. Logic*, 49(1):51–67, 2010.

[3] Valentina S. Harizanov. Pure computable model theory. In *Handbook of recursive mathematics, Vol. 1*, volume 138 of *Stud. Logic Found. Math.*, pages 3–114. North-Holland, Amsterdam, 1998.

[4] Denis R. Hirschfeldt and Walker M. White. Realizing levels of the hyperarithmetic hierarchy as degree spectra of relations on computable structures. *Notre Dame J. Formal Logic*, 43(1):51–64 (2003), 2002.

[5] David Marker. Non Σ_n axiomatizable almost strongly minimal theories. *J. Symbolic Logic*, 54(3):921–927, 1989.

[6] Hartley Rogers, Jr. *Theory of recursive functions and effective computability*. McGraw-Hill Book Co., New York, 1967.

[7] Gerald E. Sacks. *Higher Recursion Theory*. Springer-Verlag, 1990.

[8] Robert I. Soare. *Recursively Enumerable Sets and Degrees*. Perspectives in Mathematical Logic. Springer-Verlag, 1987.

[9] Walker McMillan White. *Characterizations for computable structures*. PhD thesis, Cornell University, 2000.

Department of Pure Mathematics, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1

E-mail address: csima@math.uwaterloo.ca

Department of Mathematics, 196 Auditorium Road, University of Connecticut, U-3009, Storrs, CT 06269-3009, USA

E-mail address: johanna.franklin@uconn.edu

310 Malott Hall, Cornell University, Ithaca, NY 14853-4201, USA

E-mail address: shore@math.cornell.edu