A computably stable structure with no Scott family of finitary formulas

Peter Cholak
Department of Mathematics, University of Notre Dame
Richard A. Shore
Department of Mathematics, Cornell University
Reed Solomon
Department of Mathematics, University of Connecticut

November 1, 2004

1 Introduction

One of the goals of computability theory is to find syntactic equivalences for computational properties. The Limit Lemma is a classic example of this type of equivalence: $X \subseteq \omega$ is computable from $0'$ if and only if it is arithmetically definable by a Δ^0_2 formula. A more relevant example for this paper was proved independently by Ash, Knight, Manasse and Slaman [1] and by Chishom [2]: a computable structure is relatively computably categorical if and only if it has a computably enumerable Scott family of finitary existential formulas. (These terms are defined below.) Our main theorem is a negative result which states that there is a computably stable rigid graph which does not have a Scott family of finitary formulas. Therefore, any attempt to modify the syntactic characterization of relative computably categoricity to capture computable stability using notions such as a Scott family must involve an infinitary language. In this section, we give the background definitions and motivation for this theorem. In the second section, we construct a countable family of sets $\mathcal{A}$ with a specific list of enumeration properties. In the last section, we code $\mathcal{A}$ into a rigid graph $\mathcal{G}$, prove a partial quantifier elimination theorem for $\mathcal{G}$, and use the enumeration properties of $\mathcal{A}$ and the quantifier elimination result for $\mathcal{G}$ to show that $\mathcal{G}$ does not have a Scott family of finitary formulas.

Let $\mathcal{M}$ be a countable structure in a computable language whose domain $|\mathcal{M}|$ is a subset of ω. The *degree* of $\mathcal{M}$ (denoted $\deg(\mathcal{M})$) is the Turing degree of the atomic diagram of

Cholak's research was partially supported by NSF Grants DMS 99-88716 and DMS 02-45167. Shore's research was partial supported by NSF Grant DMS-0100035. Solomon's research was partially supported by an NSF Postdoctoral Fellowship.

$(\mathcal{M}, x)_{x \in |\mathcal{M}|}$. In particular, if the language is finite, then $\mathcal{M}$ is *computable* if and only if $|\mathcal{M}|$ is computable and the interpretations of the finitely many functions and relations are all computable. We assume that every structure in this paper is coded into the natural numbers so that we can refer to its degree.

A computable structure $\mathcal{M}$ is *computably categorical* if for every computable structure $\mathcal{N} \cong \mathcal{M}$, there is a computable isomorphism between $\mathcal{M}$ and $\mathcal{N}$. For example, let $\mathcal{M}$ be any computable copy of the linear order $(\mathbb{Q}, <)$. Because the standard back and forth argument for a countable dense linear order without endpoints is effective, there is a computable isomorphism between $\mathcal{M}$ and any other computable copy of $(\mathbb{Q}, <)$. We can actually say more about this example. Suppose $\mathcal{N}$ is a copy of $(\mathbb{Q}, <)$ which is not necessarily computable. To construct an isomorphism between $\mathcal{M}$ and $\mathcal{N}$ using the back and forth method, we need to be able to calculate $|\mathcal{M}|$, $|\mathcal{N}|$, $<^{\mathcal{M}}$ and $<^{\mathcal{N}}$. Each of these is computable in $\deg(\mathcal{N})$, so there is an isomorphism between $\mathcal{M}$ and $\mathcal{N}$ which is computable in $\deg(\mathcal{N})$. That is, $\mathcal{M}$ is a computable structure such that for any (not necessarily computable) $\mathcal{N} \cong \mathcal{M}$, there is an isomorphism between $\mathcal{M}$ and $\mathcal{N}$ computable in $\deg(\mathcal{N})$. A structure with this property is called *relatively computably categorical.*

Continuing with this example, let $\mathcal{M}$ and $\mathcal{N}$ be computable copies of $(\mathbb{Q}, <)$. Because there are 2^{ω} many isomorphisms between $\mathcal{M}$ and $\mathcal{N}$ and only countably many computable functions, there must be a noncomputable isomorphism between $\mathcal{M}$ and $\mathcal{N}$. Contrast this with the situation of two computable copies of $(\omega, <, S)$ where $<$ is the usual order on ω and S is the successor function. These structures are rigid, so not only are they computably isomorphic, but every isomorphism between them is computable. If $\mathcal{M}$ is a computable structure such that for every computable $\mathcal{N} \cong \mathcal{M}$, every isomorphism between $\mathcal{M}$ and $\mathcal{N}$ is computable, then $\mathcal{M}$ is called *computably stable.* Any rigid computably categorical structure is computable stable. However, not all computable stable structures are rigid; for example, a computable copy of a two dimensional vector space over $\mathcal{Q}$ is computably stable but not rigid.

It follows from the definition of relative computable categoricity that every relatively computably categorical structure is computably categorical. Somewhat surprisingly, these notions do not coincide. Goncharov [5] used work of Selivanov [11] to construct a computable structure which is computably categorical but not relatively so. More recently, Cholak, Goncharov, Khoussainov and Shore [3] and Hirschfeldt, Khoussainov and Shore [7] constructed examples of computably categorical structures which do not remain computably categorical when the language is expanded by a single constant. None of these structures are relatively computably categorical. It is known, however, that adding extra decidability conditions will make these notions coincide. Goncharov [4] showed that any computably categorical structure for which the two quantifier diagram is decidable is relatively computably categorical. Kudinov [9] proved that Goncharov's example is optimal in terms of the level of decidability by showing there is a computably categorical structure whose one quantifier diagram is decidable but which is not relatively computably categorical.

Ash, Knight, Manasse and Slaman [1] and Chisholm [2] independently discovered a syntactical condition which is equivalent to relative computable categoricity. Their condition is expressed in terms of the existence of a special kind of Scott family (defined below). Every countable structure has a Scott family with infinitary formulas, but they showed that the relatively computably categorical structures are exactly those computable structures which have a particularly simple Scott family.

Definition 1.1. Let $\mathcal{M}$ be a countable structure. A *Scott family* for $\mathcal{M}$ is a set S of (possibly infinitary) formulas such that

1. there is a finite tuple $\overline{c}$ of elements of $\mathcal{M}$ such that the parameters occurring in each $\psi(\overline{x}) \in S$ come from $\overline{c}$,

2. for each tuple $\overline{a}$ of distinct elements of $\mathcal{M}$, there is a formula $\psi(\overline{x}) \in S$ which is satisfied by $\overline{a}$, and

3. if the tuples $\overline{a}$ and $\overline{b}$ from $\mathcal{M}$ both satisfy the same formula $\psi(\overline{x})$ from S, then there is an automorphism of $\mathcal{M}$ taking $\overline{a}$ to $\overline{b}$.

If we strengthen the third condition to say that the formulas in S define each tuple in M, then S is called a *defining family* for $\mathcal{M}$. Notice that if $\mathcal{M}$ is rigid, then any Scott family is a defining family.

Theorem 1.2 (Ash, Knight, Manasse and Slaman; Chisholm). *A computable structure is relatively computably categorical if and only if it has a Scott family S such that S is a computably enumerable set of finitary existential formulas.*

Because relative computable categoricity and computable categoricity do not coincide, it is reasonable to ask if there is a similar condition which is equivalent to the unrelativized form of computable categoricity. The condition for relatively computably categorical structures has three important features: first, it involves only finitary formulas; second, it bounds the number of quantifier alternations in these formulas (specifically, the formulas are Σ_1 in the language of the model with the parameters $\overline{c}$); and third, it bounds the arithmetic complexity of the set of these formulas (specifically, the set of these formulas is computably enumerable). Two questions immediately arise. Does every computably categorical structure have a Scott family of finitary formulas? If so, are there bounds either on the complexity of the formulas in the family or on the complexity of the set of formulas?

The Main Theorem states that even in the more restricted class of computably stable models, there is a computable graph which has no Scott family of finitary formulas. Furthermore, this graph is rigid and hence the notions of defining family and Scott family coincide for it.

Main Theorem. *There is a rigid computably stable graph G such that G has no Scott family of finitary formulas.*

The proof of this theorem splits into three steps. First, we construct a family $\mathcal{A}$ of c.e. subsets of ω with certain enumeration properties. Second, we code $\mathcal{A}$ into a graph $\mathcal{G}$ and prove a partial quantifier elimination result for $\mathcal{G}$. Third, we use the enumeration properties of $\mathcal{A}$ to show that $\mathcal{G}$ is computably categorical but cannot have a Scott family of finitary formulas.

Our construction is closely related to the work of Cholak, Goncharov, Khoussainov and Shore [3] and Hirschfeldt, Khoussainov and Shore [7]. In these papers, the authors give examples of computably categorical structures which do not remain computably categorical when a constant is added to the language. No such structure can be relatively computably categorical. To see this, suppose that $\mathcal{M}$ is relatively computably categorical but $\mathcal{M}$ is not computably categorical when some element $d \in |\mathcal{M}|$ is named. Then $\mathcal{M}$ has a c.e. Scott family of finitary existential formulas with a tuple $\overline{c}$ of parameters. $\mathcal{M}$ still has such a Scott family when we add the parameter d to the language in addition to $\overline{c}$. Therefore, $(\mathcal{M}, d)$ is relatively computably categorical but not computable categorical. This contradiction shows that $\mathcal{M}$ cannot have a c.e. Scott family of finitary existential formulas.

We strongly suspect that the examples constructed in [3] and [7] satisfy a partial quantifier elimination result similar to the one proved here for the graph $\mathcal{G}$. An appropriate result of this form would show that these examples also do not have Scott families of finitary formulas (of any complexity). Because these examples satisfy additional properties (namely, the computably dimension changes with the addition of a constant), they are more complicated than our example, and hence the quantifier elimination is more difficult to prove. Rather than proving such a result for one of these structures, we construct a new example for which it is easier to get the partial quantifier elimination. In particular, we separate the computability theory aspects (constructing of a family of sets with certain enumeration properties) from the model theory aspects (coding of the family of sets and proving partial quantifier elimination). This allows us to prove the quantifier elimination without referring to the specifics of the construction of the family of sets.

A byproduct of the Main Theorem is that $\mathcal{G}$ is not a prime model for its theory in any expansion by a finite number of constants. (The theory of $\mathcal{G}$ does have a prime model which is formed by removing the infinite components of the graph.) The following well known lemma illustrates the connection between prime models and the existence of Scott families of finitary formulas. We sketch the proof and refer the reader to Section 4.2 of Marker [10] for background on countable prime, atomic and homogeneous models.

Lemma 1.3. *Let $\mathcal{L}$ be a countable language and let $\mathcal{M}$ be a countably infinite $\mathcal{L}$ structure.*

1. *If $\mathcal{M}$ is a prime model of its theory, then $\mathcal{M}$ has a Scott family of finitary formulas without paramenters.*

2. *If $\mathcal{M}$ has a Scott family of finitary formulas with parameters $\overline{c}$, then $(\mathcal{M}, \overline{c})$ is a prime model of its theory.*

Proof. To prove the first statement, notice that $\mathcal{M}$ is also atomic and homogeneous. Because $\mathcal{M}$ is atomic, for any tuple $\overline{a}$ in $\mathcal{M}$, there is a formula $\varphi_{\overline{a}}(\overline{x})$ which generates the type of $\overline{a}$. If $\mathcal{M} \models \varphi_{\overline{a}}(\overline{b})$, then the types of $\overline{a}$ and $\overline{b}$ are the same. Because $\mathcal{M}$ is homogeneous, the map $\overline{a} \mapsto \overline{b}$ extends to an automorphism of $\mathcal{M}$. Therefore, the set of such formulas $\varphi_{\overline{a}}(\overline{x})$ is a Scott family of finitary formulas for $\mathcal{M}$ with no parameters.

To prove the second statement, it suffices to show that $(\mathcal{M}, \overline{c})$ is atomic. Fix any tuple $\overline{a}$ and let $\varphi_{\overline{a}}(\overline{x}, \overline{c})$ be a formula in the Scott family such that $\mathcal{M} \models \varphi_{\overline{a}}(\overline{a}, \overline{c})$. We claim that $\varphi_{\overline{a}}(\overline{x}, \overline{c})$ generates the type of $\overline{a}$ in $(\mathcal{M}, \overline{c})$. Fix any $\psi(\overline{x}, \overline{c})$ such that $\mathcal{M} \models \psi(\overline{a}, \overline{c})$.

Suppose for a contradiction that $\mathcal{M} \not\models \varphi_{\overline{a}}(\overline{x}, \overline{c}) \rightarrow \psi(\overline{x}, \overline{c})$. Then there is a tuple $\overline{b}$ such that $\mathcal{M} \models \varphi_{\overline{a}}(\overline{b}, \overline{c}) \wedge \neg\psi(\overline{b}, \overline{c})$. However, $\mathcal{M} \models \varphi_{\overline{a}}(\overline{b}, \overline{c})$ means that there is an automorphism of $\mathcal{M}$ sending $\overline{a}$ to $\overline{b}$. This contradicts the fact that $\mathcal{M} \models \psi(\overline{a}, \overline{c})$ and $\mathcal{M} \models \neg\psi(\overline{b}, \overline{c})$. Therefore, $(\mathcal{M}, \overline{c})$ is atomic and hence prime. □

Because of this lemma, it is possible to view our Main Theorem from a slightly different model theoretic point of view. Our graph $\mathcal{G}$ is an example of a computably categorical structure which is not the prime model of its theory in any finite expansion by constants.

Our notation follows Soare [12] for computability theory and Hodges [8] for model theory. We use φ_e to denote the e^{th} partial computable function and we assume that if $\varphi_{e,s}(n)$ converges, then both e and n are $\leq s$. In particular, for any s, only finitely many functions $\varphi_{e,s}$ have nonempty domain and we can effectively determine these domains. We also fix a computable pairing function denoted $\langle n, m \rangle$.

2 An enumeration theorem

To each partial computable function φ_e, we associate a countable family $\mathcal{B}^e$ of computably enumerable (c.e.) sets defined by $\mathcal{B}^e = \{B_i^e | i \in \omega\}$, where $n \in B_i^e$ if and only if $\langle n, i \rangle$ is in the domain of φ_e. Each of these families of sets is uniformly enumerated as $\mathcal{B}_s^e = \{B_{i,s}^e | i \in \omega\}$, where $n \in B_{i,s}^e$ if and only if $\varphi_{e,s}(\langle n, i \rangle)$ converges. We say that φ_e is a *c.e. enumeration* of $\mathcal{B}^e$. If $i \neq j$ implies that $B_i^e \neq B_j^e$, then we say the enumeration φ_e is *one-to-one*. If $\mathcal{A}$ is a countable collection of subsets of ω, then we say $\mathcal{A}$ has a *(one-to-one) c.e. enumeration* if $\mathcal{A} = \mathcal{B}^e$ for some (one-to-one) φ_e.

Let $\mathcal{A}$ be a fixed countable family of sets which has a c.e. enumeration. There is a natural notion of reducibility on the collection of all c.e. enumerations of $\mathcal{A}$. If φ_e and φ_i both enumerate $\mathcal{A}$, then we say the enumeration φ_e is *reducible to* the enumeration φ_i, denoted $\varphi_e \leq \varphi_i$, if there is a computable function f such that for all m, $B_m^e = B_{f(m)}^i$. That is, there is a computable translation from the φ_e indices for sets in $\mathcal{A}$ to the φ_i indices for these sets. We say φ_e and φ_i enumerate the same family of sets *up to computable equivalence*, denoted $\varphi_e \equiv \varphi_i$, if $\varphi_e \leq \varphi_i$ and $\varphi_i \leq \varphi_e$.

Lemma 2.1. *If φ_e and φ_i are one-to-one c.e. enumerations of the same family of sets and $\varphi_e \leq \varphi_i$, then $\varphi_e \equiv \varphi_i$.*

Proof. Since both enumerations are one-to-one, any reduction f of φ_e to φ_i must be a computable permutation of ω. Hence, f^{-1} is a computable reduction of φ_i to φ_e. □

Theorem 2.2. *There is a partial computable function ψ and a family of c.e. sets $\mathcal{A}$ with the following properties.*

1. *The function ψ gives a one-to-one enumeration of $\mathcal{A}$ which we denote by $\{A_i | i \in \omega\}$. (That is, $n \in A_i$ if and only if $\langle n, i \rangle$ is in the domain of ψ.)*

2. *$\mathcal{A}$ has a single one-to-one c.e. enumeration up to computable equivalence.*

3. *Under the enumeration given by ψ, $\exists^\infty d \exists^\infty s \exists m \neq d (A_{d,s} \subseteq A_{m,s+1})$. For each such d, s and m, we say that A_d is dumped into A_m at stage s. Furthermore, for each of the infinitely many indices d satisfying this condition, the following properties hold.*
   (a) *A_d is infinite.*
   (b) *For every k, there is a stage s_k such that for all stages $t > s_k$ and for all indices $m \neq d$, if $A_{d,t} \subseteq A_{m,t+1}$, then $A_{m,t}$ contains no numbers $\leq k$.*
   (c) *For all indices z, there is a stage s_z such that for all stages $t > s_z$ and all indices $m \neq d$, if $A_{d,t} \subseteq A_{m,t+1}$, then $m > z$.*
   (d) *For all $m \neq d$, if there is a stage s such that we enumerate the elements of $A_{d,s}$ into $A_{m,s+1}$ to cause $A_{d,s} \subseteq A_{m,s+1}$, then $A_{m,s+1} = A_m$.*

The rest of this section is devoted to a proof of Theorem 2.2. Since $\mathcal{A}$ is determined by ψ, we only need to construct ψ in an effective manner, subject to the global requirement that it gives a one-to-one enumeration of sets. Most of the time, rather than refering directly to ψ, we describe the uniform enumeration of the sets A_i for $i \in \omega$.

We insure conditions 2 and 3 of the theorem by meeting a list of requirements R_e for $e \in \omega$. Each R_e has two parts. First, if φ_e is a one-to-one enumeration of $\mathcal{A}$, then $\psi \leq \varphi_e$. This guarantees condition 2 because if φ_e is one-to-one and $\psi \leq \varphi_e$, then by condition 1 and Lemma 2.1, $\psi \equiv \varphi_e$. Second, if φ_e is a one-to-one enumeration of $\mathcal{A}$, then there is an index d_e for which there are infinitely many stages s at which there is an index $m \neq d_e$ such that $A_{d_e,s} \subseteq A_{m,s+1}$. Since there are infinitely many indices for one-to-one c.e. enumerations of $\mathcal{A}$ and since we guarantee that $d_e \neq d_i$ when $e \neq i$, there will be infinitely many indices d as in condition 3. The four properties stated in condition 3 will be clear once the construction is specified. They are the technical properties required for the proof of the Main Theorem in Section 3.

Before giving the full construction, we indicate how to satisfy one and then two requirements. We begin with a computable sequence of elements $a_0 < a_1 < \cdots$ such that $\{a_n | n \in \omega\}$ is infinite and coinfinite, and we set $A_{n,0} = \{a_n\}$ for all $n \in \omega$. At each stage s, $A_{n,s}$ will have a marking number which is only contained in $A_{n,s}$. The marking number for A_n may later enter another set, at which time we will pick a new marking number from the complement of $\{a_n | n \in \omega\}$ and enumerate it into A_n. The marking numbers allow us to match up the sets in $\mathcal{A}$ with the sets in $\mathcal{B}^e$.

R_0 begins by picking two parameters d_0 and r_0 with $d_0 < r_0$. A_{d_0} is called the *dumping set* and A_{r_0} is called the *receiving set.* We wait to find a suitable image for A_{d_0} in $\mathcal{B}^0$. That is, we wait for a stage s at which there is a unique index k such that $a_{d_0} \in B^0_{k,s}$ and for this unique index $B^0_{k,s} = A_{d_0,s}$. If we never find such a stage and index, then we never enumerate anything else into $\mathcal{A}$ and we win R_0 because $\mathcal{B}^0$ is not the same family as $\mathcal{A}$. If we find such an index, we set e_0 equal to this index. (To improve readability in the following description of the construction, we omit the subscript indicating the stage number unless it is necessary to avoid confusion. As the construction proceeds, we always compare sets as they look at the current stage.)

Once e_0 is defined, we begin the main loop for R_0. We wait for a stage at which for every $n \leq r_0$ with $n \neq d_0$, there is a unique index k_n such that $B^0_{k_n}$ contains a_n. Furthermore, we check that for these indices we have $A_n = B^0_{k_n}$. If this condition is never satisfied, then we win R_0 because $\mathcal{B}^0$ is not the same family as $\mathcal{A}$. If this condition is satisfied at stage s, we say the length of agreement l_0 is r_0. We enumerate all the elements of A_{d_0} into A_{r_0} (that is, we *dump* A_{d_0} into A_{r_0}) so that $A_{d_0,s} \subset A_{r_0,s+1}$, and we add a new marking number (from the complement of $\{a_n | n \in \omega\}$) to $A_{d_0,s+1}$. Notice that A_{r_0} is still the only set containing a_{r_0}, so we do not need to choose a new marking number for A_{r_0}. After this stage, no set A_n with $n \leq r_0$ and $n \neq d_0$ will change. In particular, their marking numbers will not change so if φ_0 is a one-to-one enumeration of $\mathcal{A}$, it must be the case that $A_n = B^0_{k_n}$ at the end of the construction. We finish this cycle for R_0 by redefining r_0 to be large and repeating the action of this paragraph.

If we ever get stuck waiting forever for the length of agreement to grow, then $\mathcal{B}^0$ is not the same family as $\mathcal{A}$ and we win R_0. If we repeat the action in the previous paragraph infinitely often, then we dump A_{d_0} into infinitely many sets and we make A_{d_0} infinite as required. Furthermore, if φ_0 is a one-to-one enumeration of $\mathcal{A}$, then we can define the computable reduction from $\mathcal{A}$ indices to $\mathcal{B}^0$ indices as follows. For any $n \neq d_0$, let s be the first stage at which $r_0 \geq n$. Before the next length of agreement, there must be a unique set $B^0_{k_n}$ which contains a_n and this set must satisfy $B^0_{k_n} = A_n$. In particular, $k_n \neq e_0$. Since the marking number for A_n never changes, A_n is the unique $\mathcal{A}$ set containing a_n. Therefore, $B^0_{k_n} = A_n$ at the end of the construction so the reduction maps n to k_n . Finally, since no A_n with $n \neq d_0$ is matched with $B^0_{e_0}$, it must be that $A_{d_0} = B^0_{e_0}$ so the reduction can map d_0 to e_0.

Before showing how R_1 interacts with R_0, we make two observations. First, A_{d_0} can be distinguished from the other sets A_n which contain a_{d_0} by the fact that it does not contain any other number of the form a_n. In the full construction, we will use this property to define e_0; e_0 will be the least index such that $B^0_{e_0}$ contains a_{d_0} and no other number of the form a_n. Second, R_0 has two possible outcomes: a waiting outcome and an infinitary outcome. Therefore, in the full construction, the tree of strategies will be the binary branching tree of finite sequences from $\{\infty, w\}$ with ∞ to the left of w. R_0 takes the ∞ outcome each time the length of agreement grows (that is, at expansionary stages) and takes the w outcome at all other stages.

The R_1 strategy which sits below the w outcome for R_0 picks its witnesses d_1 and r_1 greater than the current value of r_0. It then acts exactly like R_0 except that this strategy for R_1 is initialized at each R_0-expansionary stage. Because this R_1 strategy always works with sets whose indices are greater than those being currently considered by R_0, it does not interfere with the action of R_0.

The R_1 strategy which sits below the ∞ outcome for R_0 must act in such a way that it does not disturb R_0. When it is first eligible to act, it chooses large indices $d_1 < r_1$ and then waits for the R_0 length of agreement to be longer than r_1. At this stage, it begins to act exactly like R_0. That is, it waits until e_1 is defined as the least index such that $B^1_{e_1}$ contains a_{d_1} and no other a_n numbers. Then, it waits for its length of agreement to grow to r_1. At this stage, it dumps A_{d_1} into A_{r_1} and proceeds exactly as R_0 does when it dumps a set.

Consider how this action affects R_0. R_0 has already matched up A_{d_1} and A_{r_1} with $\mathcal{B}^0$ sets using the marking numbers. Since the marking number for A_{r_1} does not change after the dumping, it must remain matched up with the same set in $\mathcal{B}^0$. The marking number for A_{d_1}

does change, but the only $\mathcal{A}$ sets which contain a_{d_1} are A_{d_1} and A_{r_1}. Because the matching for A_{r_1} stays the same, the matching for A_{d_1} must stay the same unless $\mathcal{B}^0$ enumerates a_{d_1} into another set. However, if this occurs, then a_{d_1} is contained in three $\mathcal{B}^0$ sets but only two $\mathcal{A}$ sets. In this case, R_0 can win by never returning to the ∞ outcome. Also, R_0 can recognize this scenario by checking whether any element from A_{d_1} is contained in more $\mathcal{B}^0$ sets than $\mathcal{A}$ sets.

Finally, we need to consider how the dumping by R_0 affects this R_1 strategy. Each time R_0 takes the ∞ outcome, it adds a new marking number to A_{d_0}. If the enumeration of $\mathcal{B}^1$ is slow, it might be that R_1 never sees an expansionary stage because the numbers from A_{d_0} enter $\mathcal{B}^1$ sets only after they have been replaced as marking numbers. In this case, there might never be a stage s at which R_1 sees the current marking number for A_{d_0} in a $\mathcal{B}^1$ set even if $\mathcal{B}^1$ is the same family as $\mathcal{A}$. To fix this problem, we let R_1 guess at which $\mathcal{B}^1$ set to match with A_{d_0} by defining a parameter p_0^1 to be the least index k such that B_k^1 contains a_{d_0} and no other number of the form a_n. If φ_1 is a one-to-one enumeration of $\mathcal{A}$, then this parameter must eventually stabilize and we must have $A_{d_0} = B^1_{p_0^1}$ at the end of the construction. When R_1 is measuring its length of agreement, it does not look for a match for A_{d_0} and assumes that this set will match with $B^1_{p_0^1}$.

The full construction uses a binary branching tree of strategies with the outcomes labeled $\infty <_L w$. The nodes at level e in the tree work on requirement R_e. To simplify notation, we let φ_α and $\mathcal{B}^\alpha$ denote the partial computable function φ_e and the family of sets $\mathcal{B}^e$ respectively, for $e = |\alpha|$.

We begin the construction by fixing a computable infinite and coinfinite set of numbers $a_0 < a_1 < a_2 < \cdots$ and we declare that $A_{n,0} = \{a_n\}$. At each stage, A_n will contain a current marking number which is only contained in A_n at that stage. At the beginning of the construction, the marking number for A_n is a_n. The marking number may change as the construction proceeds and any marking number added to a set after this initial set-up is assumed to come from the complement of $\{a_n | n \in \omega\}$.

The strategy α keeps a finite list of parameters each of which may change as the construction proceeds. Technically, these parameters should be marked with stage numbers but for notational simplicity we will leave off the stage numbers unless they are needed for clarity. The parameters kept by α are: d_α (the current guess at the index of the dumping set); e_α (the current guess at the $\mathcal{B}^\alpha$ index for A_{d_α}); r_α (the $\mathcal{A}$ index for the next receiving set); p_β^α for each strategy β such that $\beta * \infty \subseteq \alpha$ (the current guess at the image of the set A_{d_β} in $\mathcal{B}^\alpha$) and l_α (the current length of agreement between our enumeration of $\mathcal{A}$ and the enumeration of $\mathcal{B}^\alpha$). If α is initialized, all of these parameters become undefined. Any parameters which are not explicitly redefined or initialized between stages retain their values at the next stage.

At stage $s+1$, we let the empty string be the first strategy which is eligible to act. We proceed as follows for each strategy α which is eligible to act until we reach a strategy of length s. When that strategy is done acting, we end the stage and initialize all strategies to the right of that strategy. If α is eligible to act at stage $s+1$, then it follows these steps.

1. If α has never acted before or if α has been initialized since it last acted, define d_α and r_α large with $d_\alpha < r_\alpha$ and define $l_\alpha = 0$. Notice that we have $A_{d_\alpha} = \{a_{d_\alpha}\}$ and $A_{r_\alpha} = \{a_{r_\alpha}\}$. Proceed to step 2.

2. We attempt to define e_α. Check if there is an index k such that only number of the form a_n contained in B^α_k is a_{d_α}. If not, then let e_α be undefined and let $\alpha * w$ be eligible to act. If so, let e_α be the least such k and proceed to step 3.

3. We attempt to define each of the parameters p^α_β for β such that $\beta * \infty \subseteq \alpha$. For each such β, in order of their priority, perform the following calculation until all such β are considered. Check if there is an index k such that the only element of the form a_n contained in B^α_k is a_{d_β}. If not, then let p^α_β be undefined and let $\alpha * w$ be eligible to act. If so, then let p^α_β be the least such index and consider the next strategy β such that $\beta * \infty \subseteq \alpha$. If all p^α_β are defined, then go to step 4.

4. We check whether we have the appropriate agreement of $\mathcal{A}$ and $\mathcal{B}^\alpha$ up to r_α. Check whether the following conditions are met at this stage.

   (a) $r_\alpha < l_\beta$ for all β such that $\beta * \infty \subseteq \alpha$.

   (b) For all $i \leq r_\alpha$ such that $i \neq d_\alpha$ and $i \neq d_\beta$ for any β such that $\beta * \infty \subseteq \alpha$, there is a unique index k_i such that $B^\alpha_{k_i}$ contains the current marking number for A_i. Furthermore, for this index k_i, check that we currently have $A_i = B^\alpha_{k_i}$.

   (c) For every n, if $n \in A_{d_\delta}$ for some strategy δ such that $\alpha * \infty \subseteq \delta$ and n is in exactly m many different $\mathcal{A}$ sets, then n is in at most m many $\mathcal{B}^\alpha$ sets.

   If any of these conditions fail, then let $\alpha * w$ be eligible to act. If all of these conditions are met, then act as follows.

   (i) Put all of the numbers currently in A_{d_α} into A_{r_α}.

   (ii) Choose a new large marking number for A_{d_α} from the complement of $\{a_n | n \in \omega\}$ and enumerate it into A_{d_α}. (The number a_{r_α} remains the marking number for A_{r_α}.)

   (iii) Define $l_{\alpha,s+1}$ to be equal to $r_{\alpha,s}$ and define $r_{\alpha,s+1}$ to be large. Let $\alpha * \infty$ be eligible to act.

This completes the description of the action for α. As usual, we say that α is on the true path if α is the left-most node at level $|\alpha|$ in the tree of strategies which is eligible to act infinitely often.

Lemma 2.3. *Let α be on the true path. If $\alpha * \infty$ is on the true path, then the limit of $l_{\alpha,s}$ as $s \to \infty$ is ∞.*

Proof. Let s be the least stage at which α is eligible to act and after which α is never initialized. The value of l_α is set to 0 at stage s and l_α is never initialized after this stage. Therefore, l_α can only change values because of the action in step 4 by α. Because $\alpha * \infty$ is on the true path, there are infinitely many stages after s at which α meets the conditions of step 4 and $\alpha * \infty$ is eligible to act. Let $t_0 < t_1 < t_2 < \cdots$ denote these stages. At stage t_n, l_{α,t_n} is set to r_{α,t_n-1} and r_{α,t_n} is defined to be large. Since r_α is never initialized after stage s, $r_{\alpha,t_{n+1}-1}$ will have the same value as r_{α,t_n}. Hence, when $l_{\alpha,t_{n+1}}$ is defined to be equal to $r_{\alpha,t_{n+1}-1}$ we have

$$l_{\alpha,t_{n+1}} = r_{\alpha,t_{n+1}-1} = r_{\alpha,t_n} > r_{\alpha,t_n-1} = l_{\alpha,t_n}.$$

Therefore, $l_{\alpha,t_{n+1}} > l_{\alpha,t_n}$ for all n and hence l_α approaches ∞. $\square$

Lemma 2.4. *Let α be on the true path. If $\alpha * w$ is on the true path, then φ_α is not a 1-1 enumeration of $\mathcal{A}$.*

Proof. Let s be the first stage at which α is eligible to act after it is initialized for the last time. The parameter d_α is defined at stage s and its value is never changed after this stage. At this stage $A_{d_\alpha} = \{a_{d_\alpha}\}$ and a_{d_α} will never appear in another A_n unless α puts it there by its actions in step 4. Furthermore, the only numbers which can enter A_{d_α} after this stage are new marking numbers put into A_{d_α} by α in step 4. Therefore, at the end of the construction, the only number in A_{d_α} of the form a_n is a_{d_α}. (Of course, similar comments hold for any strategy β with $\beta \subseteq \alpha$ since these strategies are also on the true path and will not be initialized after stage s.) The parameter r_α is also defined at this stage and it will remain constant unless redefined by α in step 4. Because $\alpha * w$ is on the true path, there is a stage $s' > s$ after which $\alpha * \infty$ is never eligible to act. We consider various sequences of events which could occur after stage s'. In each case, we show why φ_α is not a 1-1 enumeration of $\mathcal{A}$.

Suppose there is a stage $t > s'$ after which the parameter e_α is never defined. In this situation, either there is no set B^α_k which contains a_{d_α} or else every set B^α_k which contains a_{d_α} also contain some other number of the form a_n. In either case, $\mathcal{B}^\alpha$ has no set which corresponds to A_{d_α}, so $\mathcal{B}^\alpha$ and $\mathcal{A}$ are different families of sets and we are done. Therefore, we assume that for every t, there is a stage $u > t$ at which e_α is defined.

It is also possible that e_α changes its value infinitely often. However, if e_α changes value, it either becomes smaller since we find a smaller index satisfying the condition in step 2 or it must change because a number a_n other than a_{d_α} enters $B^\alpha_{e_\alpha}$. In the latter case, e_α can never return to its previous value. Hence, if e_α is equal to m, then it can only change finitely often without all of its future defined values being larger than m. Therefore, if e_α changes values infinitely often, then the limit of these values is infinite. This means that every set B^α_k which contains a_{d_α} must eventually contain some other number of the form a_n. As above, $\mathcal{B}^\alpha$ and $\mathcal{A}$ are different families of sets and we are done. Therefore, we assume there is a stage after which α is never initialized, $\alpha * \infty$ is never eligible to act, the value of e_α is defined and constant, and α always passes to step 3 when it is eligible to act.

After this stage (by the comments in the first paragraph of this proof), the parameters d_β for β such that $\beta * \infty \subseteq \alpha$ are fixed and the only number of the form a_n ever to appear in A_{d_β} is a_{d_β}. We now argue in a similar fashion as above that each of the parameters p^α_β must eventually be defined and constant. Suppose that there is a stage after which one of the parameters p^α_β is never defined. In this situation, either there is no set B^α_k which contains a_{d_α} or else every set B^α_k which contains a_{d_α} also contains another number of the form a_n. In either case, $\mathcal{B}^\alpha$ and $\mathcal{A}$ are different families of sets and we are done. Suppose that p^α_β is redefined infinitely often. When p^α_β changes values, it either gets smaller since we have found a smaller index satisfying the conditions of step 3 or it changes because some number of the form a_n enters $B^\alpha_{p^\alpha_\beta}$. Hence, if p^α_β is equal to m, then it can only be redefined finitely often without its values becoming permanently larger than m. Therefore, if p^α_β changes values infinitely often, these values approach infinity. However, this means that every set B^α_k which contains a_{d_β} must contain some other number of the form a_n. In this case, there is no set in $\mathcal{B}^\alpha$ corresponding to A_{d_β} and we are done. Therefore, we assume that there is a stage t after which α is never initialized, $\alpha * \infty$ is never eligible to act, the values of e_α and all the p^α_β parameters are defined and constant, and α proceeds to step 4 when it is eligible to act.

We are left to consider the conditions in step 4 after such a stage t. At every stage after t at which α is eligible to act, α proceeds to step 4 and takes outcome $\alpha * w$. Therefore, at each such stage, one of the conditions in step 4 must fail to hold. By Lemma 2.3, condition 4(a) will eventually be met for any value of r_α. Furthermore, r_α is never redefined after stage t (since such a redefinition could only happen if the conditions in step 4 were met), so once condition 4(a) is met, it will continue to be met at every future stage. Therefore, we can assume without loss of generality that condition 4(a) holds at every stage after t.

Consider condition 4(c) at stage t. No strategy δ with $\alpha * \infty \subseteq \delta$ is ever eligible to act after stage t, so the parameter d_δ is permanently fixed and the set A_{d_δ} does not change after stage t. In particular, it is never dumped into another set after this stage. Suppose that there is a number n in A_{d_δ} which occurs in more $\mathcal{B}^\alpha$ sets than in $\mathcal{A}$ sets. Either $n = a_{d_\delta}$ or n is a (possibly former) marking number for A_{d_δ}. In either case, n can only enter more $\mathcal{A}$ sets if some $\mathcal{A}$ set containing n is dumped into another $\mathcal{A}$ set. We know that A_{d_δ} is never dumped into another $\mathcal{A}$ set after stage t. Also, any other set containing n must have the form A_{r_δ} for some (possibly previous) value of r_δ. Since our parameters are always defined to be large, we never reuse a set of the form A_{r_δ} as a dumping set for any requirement. Therefore, no set containing n will ever be dumped into another $\mathcal{A}$ set and hence n can never enter another $\mathcal{A}$ set. Because φ_α has enumerated n into more sets than n enters in $\mathcal{A}$, φ_α cannot be a 1-1 enumeration of $\mathcal{A}$. Hence, if condition 4(c) fails at stage t, it will fail at every future stage and φ_α is not a 1-1 enumeration of $\mathcal{A}$.

For the final case, assume that condition 4(c) holds at stage t, so condition 4(b) must fail at every stage after t at which α is eligible to act. Consider which strategies could cause a change in a set A_i with $i \leq r_\alpha$ after stage t. No strategy to the left of α or below $\alpha * \infty$ is eligible to act after stage t. Any strategy to the right of α or below $\alpha * w$ has parameters larger than r_α, so they will not affect these sets. Therefore, only strategies β with $\beta * \infty \subseteq \alpha$ can cause any changes in the sets A_i with $i \leq r_\alpha$ after stage t. Moreover, because $l_\beta > r_\alpha$ and $r_\beta > l_\beta$, these strategies are dumping A_{d_β} into sets with indices larger than r_α. Therefore, the only sets A_i with $i \leq r_\alpha$ which can change after stage t are the sets of the form A_{d_β} and they will change only by the addition of new marking numbers. In particular, the sets with indices i considered in step 4(b) will not change after stage t and their marking numbers remain constant. If the marking number from one of these sets either does not enter any B^α_k set or enters two such sets, then the families $\mathcal{B}^\alpha$ and $\mathcal{A}$ are different. Therefore, if condition 4(b) is never met, we are done. $\square$

Lemma 2.5. *Let α be on the true path. If φ_α is a 1-1 enumeration of $\mathcal{A}$, then $\alpha * \infty$ is on the true path and d_α reaches a limit for which A_{d_α} satisfies $\exists^\infty s \exists m \neq d_\alpha (A_{d_\alpha,s} \subseteq A_{m,s+1})$. Furthermore, there is a computable function f such that for all i, $A_i = B^\alpha_{f(i)}$.*

Proof. Assume that α is on the true path and φ_α is a one-to-one enumeration of $\mathcal{A}$. By Lemma 2.4, we know that $\alpha * \infty$ is on the true path. Furthermore, once we reach a stage at which α is eligible to act and the path is never to the left of α, d_α is defined for the last time and has reached its limit. At each subsequent stage at which $\alpha * \infty$ is eligible to act, A_{d_α} is dumped into A_{r_α} and A_{d_α} gets a new marking number. Therefore, A_{d_α} satisfies $\exists^\infty s \exists m \neq d_\alpha(A_{d_\alpha,s} \subseteq A_{m,s+1})$. It remains to show that the existence of the computable function f.

Let s be a stage such that α is never initialized after s and the parameters e_α and p^α_β have reached their limits by stage s. (These limits are guaranteed by the proof of Lemma 2.4.) The set A_{d_α} is uniquely determined by the fact that it contains a_{d_α} and no other number of the form a_n. Because $B^\alpha_{e_\alpha}$ shares this property and because φ_α is a 1-1 enumeration of $\mathcal{A}$, it must be that $B^\alpha_{e_\alpha} = A_{d_\alpha}$. Similarly, for each β such that $\beta * \infty \subseteq \alpha$, we must have $B^\alpha_{p^\alpha_\beta} = A_{d_\beta}$. Therefore, we set $f(d_\alpha) = e_\alpha$ and $f(d_\beta) = p^\alpha_\beta$.

Let i be such that $i \neq d_\alpha$ and $i \neq d_\beta$ for any $\beta * \infty \subseteq \alpha$. By Lemma 2.3, we know there is a stage $t > s$ such that α is eligible to act at t, α takes outcome $\alpha * \infty$ at t and the parameter l_α is defined to be larger than i at stage t. By the conditions in step 4, there must be a unique index k_i such that the current marking number for A_i is contained in $B^\alpha_{k_i}$ at stage t. Set $f(i) = k_i$.

This completes the definition of f and it remains to check that at the end of the construction $A_i = B^\alpha_{f(i)}$. First, consider which strategies could put a number into A_i after stage t. No strategy to the left of α acts after this stage. The strategies to the right of α and below $\alpha * w$ are initialized at the end of stage t and so work with parameters larger than i. As noted in the previous proof, condition 4(a) implies that any strategy β with $\beta * \infty \subseteq \alpha$ is dumping A_{d_β} into sets with indices larger than i and is only adding marking numbers to A_{d_β}. The strategy α is also dumping A_{d_α} into sets with indices larger than i and is adding marking numbers to A_{d_α}. Therefore, the only strategy which could add numbers to A_i is a strategy δ with $\alpha * \infty \subseteq \delta$. Furthermore, this can only happen if either $i = d_\delta$ or $i = r_\delta$ at stage t. (Any parameter defined by such a δ after stage t will be larger than i.)

Suppose $i = r_\delta$ and δ adds numbers to A_i after stage t by dumping A_{d_δ} into A_i. Because δ choses a new value for r_δ when it does the dumping, this action can happen only once and then A_i never changes again. Also, the marking number for $A_i = A_{r_\delta}$ does not change after A_{d_δ} is dumped into A_{r_δ}. Therefore, in this case, the marking number for A_i is constant after stage t and A_i is the only set containing this number. Since $B^\alpha_{f(i)}$ already contains this marking number, it must be that $B^\alpha_{f(i)} = A_i$.

Suppose $i = d_\delta$. In this case, δ adds a new marking number to A_i at each stage after t at which $\delta * \infty$ is eligible to act (until δ is initialized, which may or may not happen depending on whether δ is on the true path). The strategy $\alpha * \infty$ is eligible to act at stage t and by assumption, $l_\alpha > i$. Therefore, there is a unique set B^α_b which contains the marking number m_0 for A_i at stage t. (By the definition of f, we set $f(i) = b$.) Let $t_1 < t_2 < \cdots$ be the (possibly finite) sequence of stages after t at which $\delta * \infty$ is eligible to act (and before it is initialized). Let m_j denote the marking number added to A_i by δ at stage t_j. Notice that these marking numbers are the only numbers added to A_i after t and that the only changes in A_i occur at the t_j stages.

The action at t_j can be summarized as follows.

1. At the start of t_j, m_{j-1} is the marking number for A_i and $a_{r_{\delta,t_j}}$ is the marking number for $A_{r_{\delta,t_j}}$.

2. $\alpha * \infty$ is on the current path and $l_\alpha > r_{\delta,t_j} > m_{j-1}$. (The second inequality follows by an induction explained below.) This point has two important consequences. First, there is a unique c_j such that $m_{j-1} \in B^\alpha_{c_j}$ at stage t_j and currently we have $A_i = B^\alpha_{c_j}$. Second, there is a unique p_j such that $a_{r_{\delta,t_j}} \in B^\alpha_{p_j}$ at stage t_j and currently we have $A_{r_{\delta,t_j}} = B^\alpha_{p_j}$.

3. When δ acts at t_j, it performs three notable actions. First, it dumps A_i into $A_{r_{\delta,t_j}}$, so m_{j-1} is contained in only A_i and $A_{r_{\delta,t_j}}$ among the $\mathcal{A}$ sets. Second, it adds m_j to A_i, so m_j is only contained in A_i among the $\mathcal{A}$ sets. Third, it defines r_{δ,t_j+1} to be large and in particular, larger than m_j. Since δ does not redefine its parameters again until stage t_{j+1}, we have that $r_{\delta,t_{j+1}} = r_{\delta,t_j+1} > m_j$ which explains the second inequality in 2.

To show that $A_i = B^\alpha_{f(i)}$ at the end of the construction, it suffices to show that $c_j = b$ for all $j \geq 1$. For the base case, we show that $c_1 = b$. By definition, b is the unique index such that B^α_b contains the marking number m_0 for A_i at stage t. Since A_i does not change between the action of α at t and the action of δ at t_1, we have that m_0 is still the marking number for A_i at the start of t_1. By definition, c_1 is the unique index such that $m_0 \in B^\alpha_{c_1}$ at stage t_1. Therefore, c_1 must be equal to b.

For the induction step, assume that $c_j = b$ and we show that $c_{j+1} = b$. By induction, we know that at the end of stage t_j, $A_i = B^\alpha_b = B^\alpha_{c_j}$. Furthermore, by the comments above on the action at t_j we know the following. On the $\mathcal{A}$ side, m_{j-1} is only contained in A_i and $A_{r_{\delta,t_j}}$; $a_{r_{\delta,t_j}}$ is only contained in $A_{r_{\delta,t_j}}$; and m_j is only contained in A_i. On the $\mathcal{B}^\alpha$ side, m_{j-1} is only contained in $B^\alpha_b = B^\alpha_{c_j}$ and $a_{r_{\delta,t_j}}$ is only contained in $B^\alpha_{p_j}$. Furthermore, since $A_i = B^\alpha_b$ and $A_{r_{\delta,t_j}} = B^\alpha_{p_j}$, we know that $p_j \neq b$.

At the start of stage t_{j+1}, we know that $l_\alpha > r_{\delta,t_{j+1}} = r_{\delta,t_j+1} > m_j$. Since $a_{r_{\delta,t_j}}$ is still the marking number for $A_{r_{\delta,t_j}}$ and it is already in $B^\alpha_{p_j}$, we must have that at t_{j+1}, $A_{r_{\delta,t_j}} = B^\alpha_{p_j}$ and in particular that $m_{j-1} \in B^\alpha_{p_j}$. Since m_{j-1} is in B^α_b as well and $b \neq p_j$, we have that m_{j-1} is in at least two $\mathcal{B}^\alpha$ sets. On the $\mathcal{A}$ side, m_{j-1} is only in A_i and $A_{r_{\delta,t_j}}$ and hence it is only in two $\mathcal{A}$ sets. Therefore, by condition 4(c) in step 4 for α (recall that $m_{j-1} \in A_i$, $i = d_\delta$ and $\alpha * \infty \subseteq \delta$), m_{j-1} cannot be in any other $\mathcal{B}^\alpha$ sets. By the definition of c_{j+1}, we have $A_i = B^\alpha_{c_{j+1}}$ at stage t_{j+1}, so m_{j-1} is in $B^\alpha_{c_{j+1}}$. Therefore, either $c_{j+1} = b$ or $c_{j+1} = p_j$. Since $a_{r_{\delta,t_j}} \in B^\alpha_{p_j}$ and $a_{r_{\delta,t_j}} \notin A_i$, we cannot have $c_{j+1} = p_j$. Hence, $c_{j+1} = b$ as required. $\square$

Lemma 2.6. *Each index d from condition 3 of Theorem 2.2 satisfies the four properties listed in condition 3.*

Proof. Let d be such an index. Since A_d is dumped infinitely often into other sets, d must be of the form d_α for some strategy α with $\alpha * \infty$ on the true path. Each time A_d is dumped into a receiving set, A_d gets a new marking number and the index for the next receiving set is chosen large. Because A_d gets a new marking number infinitely often, property 3(a) is satisfied. Because the new receiving index r_α is chosen large, A_{r_α} only contains a_{r_α} until A_d is dumped into A_{r_α}. For any fixed k, there is eventually a stage at which $a_{r_\alpha} > k$. After this stage, A_d is only dumped into sets which currently contain no numbers less than k. Therefore, property 3(b) is satisfied. Furthermore, for any fixed z, there is eventually a stage such that the index $r_\alpha > z$. After this stage, A_d is only dumped into receiving sets with index $> z$. Therefore, property 3(c) is satisfied. Finally, since the indices for receiving sets for any strategy are always chosen large (and hence have never been used before), indices are never reused during the construction. Therefore, if A_d is dumped into a set A_m at stage s, the index m is never used again, so A_m does not change after stage $s+1$. Therefore, property 4(d) is satisfied.

□

3 Coding sets into a graph

Let $\mathcal{A} = \{A_i | i \in \omega\}$ be the family of sets constructed in the previous section. There are many ways to code such a family into a directed graph (G, R). We give one method for which it is particularly easy to prove a partial quantifier elimination result. This quantifier elimination result is quite general and it only depends on the facts that if $i \neq j$, then $A_i \neq A_j$ and that for any n, there is a number larger than n in at least one of the sets in $\mathcal{A}$.

The directed graph (G, R) has one component for each A_i. We begin with points c_i for $i \in \omega$, called *coding locations*, which satisfy the relation $c_i R c_i$ for $i \in \omega$. The set of coding locations is definable in G as the set of points p for which pRp holds. Each set A_i is coded using cycles attached to c_i whose lengths are powers of 7. (We pad the coding by using powers of 7 to make the results below easier to prove.) If $0 \in A_i$, we add new points x_j for $1 \leq j \leq 6$ to G along with the relations $c_i R x_1$, $x_j R x_{j+1}$ for $1 \leq i \leq 5$ and $x_6 R c_i$. We call this cycle a *loop* of length 7. For each $n \in A_i$, we add a loop of length 7^{n+1} attached to c_i. (Of course, we keep these loops disjoint from each other.) Because the graph is directed and because $i \neq j$ implies $A_i \neq A_j$, G is rigid.

To measure distances in G, we do not want to distinguish between moving forwards and backwards in G. Let $\hat{R}$ denote the symmetric closure of R: $a\hat{R}b$ holds if and only if either aRb or bRa holds. If a and b are in the same component of G, then the *distance* between them is defined by $d(a, b) = n$ if there is a path from a to b in $(G, \hat{R})$ of length n and there is no path of shorter length. If a and b are in different components of G, then $d(a, b) = \infty$.

Using the distance function, we define *neighborhoods* of the form $N(\overline{x}, 7^k)$ for any tuple $\overline{x}$ in G. The most obvious definition for such a neighborhood would be the set of points within distance 7^k of any $x_i \in \overline{x}$. However, we want such neighborhoods to be finite, which conflicts which the fact that if A_i is infinite, then there are infinitely many points within 7^k of c_i. So, we define them in a slightly different way which make the induction in our argument below carry through easily.

The neighborhood $N(\overline{x}, 7^0)$ consists of the points in $\overline{x}$. For $k > 0$, we form $N(\overline{x}, 7^k)$ as follows. All the points in $\overline{x}$ are in $N(\overline{x}, 7^k)$. For each $x_i \in \overline{x}$ which is not a coding location, we proceed both forward and backward from x_i for 7^k many steps adding all of these points to $N(\overline{x}, 7^k)$, except that we stop adding points in a particular direction if we run into a coding location in that direction. (We do add any coding location we hit to $N(\overline{x}, 7^k)$.) For each coding location c we have in $N(\overline{x}, 7^k)$ so far, we add all cycles off of c which have length $\leq 7^k$. (These cycles correspond to the numbers $< k$ in the set from $\mathcal{A}$ coded by c.) We also add up to $2k$ many more "straight rays" of length 7^k attached to c as follows. For each number $i \geq k$

in the set coded by c, there is a forward ray of length 7^k proceeding from c and there is a backward ray of length 7^k receding from c. We add the maximal number, up to k many more, of forward rays and the maximal number, up to k many more, of backward rays of length 7^k from c which do not contain any element of $\overline{x}$, except possibly the coding location itself. This completes the definition of $N(\overline{x}, 7^k)$.

The definition of $N(\overline{x}, 7^k)$ does not define an actual set in G. We potentially have a choice to make when we add the forward and backward rays in the last step since there may be more that k many numbers greater than k in the set coded by c. (These are the only points about which we have any choice. The rest of the neighborhood is determine by the conditions above.) However, it does define the isomorphism type of a finite subgraph of G containing $\overline{x}$ uniquely. This subgraph can be described up to isomorphism in G by a formula $\psi(\overline{x})$ in the language of graph theory. This formula is Σ_2 since it begins with a large existential block over the elements other than $\overline{x}$ in the subgraph and then needs universal quantifiers to express that certain cycles of length $\leq 7^k$ do not exist in the subgraph and (possibly) that no more rays of length 7^k extend forward or backward from a coding location in the subgraph (if there is a coding location in the subgraph with fewer than $2k$ many additional forward and backward rays). We say that $N(\overline{x}, 7^k) \cong N(\overline{y}, 7^k)$ if these finite subgraphs are isomorphic by a map sending $\overline{x}$ to $\overline{y}$. We write $a \in N(\overline{x}, 7^k)$ to mean that there is an instantiation of this subgraph which includes the point a. What this really means is that either a is one of the points which is actually determined by the definition or else there is a coding location c determined to be in the subgraph such that $d(c, a) \leq 7^k$. (Notice that the isomorphism type of the subgraph determined by the formula $\psi(\overline{x})$ for $N(\overline{x}, 7^k)$ is finite but that there may be infinitely many points a such that $a \in N(\overline{x}, 7^k)$.) Similarly, we write $N(\overline{y}, 7^l) \subset N(\overline{x}, 7^k)$ to stand for the statement that for every point a, if $a \in N(\overline{y}, 7^l)$, then $a \in N(\overline{x}, 7^k)$.

Lemma 3.1. *The following properties hold for this definition of neighborhood.*

1. *Any neighborhood $N(\overline{x}, 7^k)$ contains only finitely many coding locations.*
2. *If c is a coding location and $c \in N(\overline{x}, 7^k)$, then $N(c, 7^k) \subseteq N(\overline{x}, 7^k)$.*
3. *If $d(a, x_i) < 7^k$ for some $x_i \in \overline{x}$, then $a \in N(\overline{x}, 7^k)$.*
4. *If $a \in N(\overline{x}, 7^k)$, then there is an $x_i \in \overline{x}$ for which $d(a, x_i) \leq 2 \cdot 7^k$.*
5. *If $d(a, x_i) \leq 4 \cdot 7^k$ for some $x_i \in \overline{x}$, then $N(a, 7^k) \subseteq N(\overline{x}, 7^{k+1})$.*
6. *If $d(a, x_i) > 4 \cdot 7^k$ for all $x_i \in \overline{x}$, then $N(a, 7^k) \cap N(\overline{x}, 7^k) = \emptyset$.*
7. *If $N(a, 7^k) \cap N(\overline{x}, 7^k) \neq \emptyset$, then $a \in N(\overline{x}, 7^{k+1})$.*

Proof. Statements 1, 2 and 3 follow immediately from the definition of neighborhood. The bound $2 \cdot 7^k$ in statement 4 arises because a could be included in the neighborhood because $d(a, c) = 7^k$ for a coding location c for which $d(c, x_i) = 7^k$. For statement 5, if $b \in N(a, 7^k)$, then $d(b, a) \leq 2 \cdot 7^k$ by statement 4. Since $d(a, x_i) \leq 4 \cdot 7^k$, we have $d(b, x_i) \leq 2 \cdot 7^k + 4 \cdot 7^k < 7^{k+1}$. For statements 6 and 7, suppose $b \in N(a, 7^k) \cap N(\overline{x}, 7^k)$. Then $d(a, b) \leq 2 \cdot 7^k$ and there is an $x_i \in \overline{x}$ for which $d(b, x_i) \leq 2 \cdot 7^k$. Therefore, $d(a, x_i) \leq 4 \cdot 7^k$. Statement 7 follows because

$d(a, x_i) < 7^{k+1}$. Statement 6 follows because $d(a, x_i) \leq 4 \cdot 7^k$ contradicts the assumption of this statement. $\square$

We now define the equivalence relations which will lead to our quantifier elimination result. For each $k \geq 0$ and each pair of tuples $\overline{x}$, $\overline{y}$ of the same length, we define $\overline{x} E_k \overline{y}$ if and only if the neighborhood $N(\overline{x}, 7^k)$ with constants for the parameters $\overline{x}$ is isomorphic to the neighborhood $N(\overline{y}, 7^k)$ with constants for the parameters $\overline{y}$.

Lemma 3.2. *Suppose that $\overline{x} E_{k+1} \overline{y}$ and a is any element of G. Then there is a b such that $\overline{x} a E_k \overline{y} b$.*

Proof. The proof proceeds by considering cases.

Case 1. Assume there exists an $x_i \in \overline{x}$ such that $d(a, x_i) \leq 4 \cdot 7^k$. Then, $N(a, 7^k) \subseteq N(\overline{x}, 7^{k+1})$. Since $N(\overline{x}, 7^{k+1}) \cong N(\overline{y}, 7^{k+1})$, there is a corresponding element b in $N(\overline{y}, 7^{k+1})$.

Case 2. Assume for all $x_i \in \overline{x}$ that $d(x_i, a) > 4 \cdot 7^k$. In this case, $N(\overline{x}, 7^k) \cap N(a, 7^k) = \emptyset$ so $N(\overline{x}a, 7^k)$ is the disjoint union of these two neighborhoods. It suffices to find b such that $N(b, 7^k) \cong N(a, 7^k)$ and $N(b, 7^k) \cap N(\overline{y}, 7^k) = \emptyset$.

Case 2(a). Assume for all $y_i \in \overline{y}$ that $d(y_i, a) > 4 \cdot 7^k$. Then $N(a, 7^k) \cap N(\overline{y}, 7^k) = \emptyset$, so we can chose $b = a$.

Case 2(b). Assume that there is a $y_i \in \overline{y}$ such that $d(y_i, a) \leq 4 \cdot 7^k$. In this case, $N(a, 7^k) \subseteq N(\overline{y}, 7^{k+1})$. We consider three possibilities for $N(a, 7^k)$.

First, assume $N(a, 7^k)$ is just a straight line of length $2 \cdot 7^k + 1$ and $N(a, 7^k)$ does not contain a coding location. Since arbitrarily large numbers occur in $\mathcal{A}$, there is a b such that $N(b, 7^k)$ is a straight line which does not contain a coding location and $N(b, 7^k) \cap N(\overline{y}, 7^k) = \emptyset$.

Second, assume that a is on a cycle of length $\leq 7^k$ attached to a coding location c. In this case, $N(a, 7^k)$ is equal to $N(c, 7^k)$. Let $d_0, d_1, \ldots$ be the (possibly finite) list of all coding locations for which $N(d_i, 7^k) \cong N(c, 7^k)$. It suffices to show that at least one of these coding locations satisfies $N(d_i, 7^k) \cap N(\overline{y}, 7^k) = \emptyset$ for then we can take the point corresponding to a in $N(d_i, 7^k)$ as the witness b. Suppose that none of these neighborhoods is disjoint from $N(\overline{y}, 7^k)$. Since each $N(d_i, 7^k)$ intersects $N(\overline{y}, 7^k)$, each coding location d_i must be in $N(\overline{y}, 7^{k+1})$ and each neighborhood $N(d_i, 7^k)$ is contained in $N(\overline{y}, 7^{k+1})$. Because $N(\overline{y}, 7^{k+1})$ can only contain finitely many coding locations, the list of coding locations d_i for which $N(d_i, 7^k) \cong N(c, 7^k)$ is finite. Let $d_0, \ldots, d_n$ denote these coding locations and note that $c = d_i$ for some $0 \leq i \leq n$. Since $N(\overline{y}, 7^{k+1}) \cong N(\overline{x}, 7^{k+1})$, there must be coding locations $e_0, \ldots, e_n$ in $N(\overline{x}, 7^{k+1})$ such that $N(e_i, 7^k) \cong N(d_i, 7^k) \cong N(c, 7^k)$. Furthermore, each $N(e_i, 7^k)$ must intersect $N(\overline{x}, 7^k)$. However, the set $\{d_0, \ldots, d_n\}$ contains all coding locations for which $N(d_i, 7^k) \cong N(c, 7^k)$. Therefore, the sets $\{d_0, \ldots, d_n\}$ and $\{e_0, \ldots, e_n\}$ are the same, $c = e_j$ for some j and $N(c, 7^k)$ intersects $N(\overline{x}, 7^k)$. Because $N(a, 7^k) = N(c, 7^k)$, $N(a, 7^k)$ intersects $N(\overline{x}, 7^k)$, contrary to the hypothesis of case 2.

Third, assume that there is a coding location c such that $d(c, a) \leq 7^k$ but a is not on a cycle of length $\leq 7^k$ attached to c. That is, a is one of the points on a straight line of length 7^k attached to c in $N(a, 7^k)$. In this case, $N(a, 7^k)$ looks just like $N(c, 7^k)$ except that the spoke from c through a will be of length $7^k + d(c, a)$ and it may be an extra spoke in addition to the ones in $N(c, 7^k)$. We can now repeat the argument of the previous paragraph, except that we work with the isomorphism type I of $N(a, 7^k)$. That is, let $(d_0, e_0), (d_1, e_1), \ldots$ be pairs of points such that the d_i are distinct coding locations and e_i is a fixed point such that $d(e_i, d_i) = d(a, c)$ and $N(e_i, 7^k) \cong N(a, 7^k)$. If there is an e_i such that $N(e_i, 7^k) \cap N(\overline{y}, 7^k) = \emptyset$, then we can use e_i as the witness b. Otherwise, if all $N(e_i, 7^k)$ intersect $N(\overline{y}, 7^k)$, then $N(d_i, 7^k) \subseteq N(e_i, 7^k) \subseteq N(\overline{y}, 7^{k+1})$ for each i. There can be only finitely many distinct coding locations d_i with this property, so the argument can proceed as in the previous paragraph. $\square$

For any fixed $k \geq 0$ and $n > 0$, we have the equivalence relation E_k on the set of n-tuples from G as defined above. By the definition of E_k, there are finitely many equivalence classes of n tuples under this relation and each of these classes can be captured by a formula in the language of graph theory. Let $\Gamma_{k,n}$ denote this finite set of formulas and let Γ be the union of all the $\Gamma_{k,n}$. We claim that every formula in the language of graph theory is equivalent in (G, R) to a boolean combination of the formulas in Γ. To prove this fact, we use the notion of a graded back-and-forth system. Since we will need a similar fact when we expand the language by a finite tuple of constants from G, we state the following definition and lemma from Hodges [8] in a form that we can apply in this expanded setting. (See Section 3.3 of Hodges for a complete discussion.)

Definition 3.3. Fix a finite tuple (possibly empty) of parameters $\overline{c}$ from G. Suppose that $E_k^{\overline{c}}$, $k \in \omega$, is any family of equivalence relations on the tuples from the expanded graph structure $(G, \overline{c})$. The relations $E_k^{\overline{c}}$, $k \in \omega$ form a *graded back-and-forth system* for $(G, \overline{c})$ if the following two properties hold.

1. If $\overline{x}E_0^{\overline{c}}\overline{y}$, then for all atomic formulas $\psi(\overline{u})$ in the language of graph theory plus constants for $\overline{c}$, $\psi(\overline{x})$ holds in $(G, \overline{c})$ if and only if $\psi(\overline{y})$ holds in $(G, \overline{c})$.
2. If $\overline{x}E_{k+1}^{\overline{c}}\overline{y}$ and a is any element of G, then there is $b \in G$ such that $\overline{x}aE_k^{\overline{c}}\overline{y}b$.

Lemma 3.4. *Suppose $E_k^{\overline{c}}$, $k \in \omega$, is a graded back-and-forth system for $(G, \overline{c})$ and for each k and n, the equivalence relation $E_k^{\overline{c}}$ has only finitely many equivalence classes among the tuples of length n in G. Furthermore, suppose that each of these equivalence classes can be defined by a formula in the language of graph theory with constants added for $\overline{c}$. Let $\Psi^{\overline{c}}$ denote the set of all such formulas. Then every formula in the language of graph theory with constants for $\overline{c}$ is equivalent to a boolean combination of formulas from $\Psi^{\overline{c}}$ over $(G, \overline{c})$.*

We show that the family of equivalence relations E_k defined above satisfies the conditions of a graded back-and-forth system when the list of parameters in empty. By definition, $\overline{x}E_0\overline{y}$ if and only if the subgraphs consisting of just the points in $\overline{x}$ and $\overline{y}$ are isomorphic. Since atomic formulas in the language of graph theory can only say whether two elements are related in the graph or not, E_0 meets condition 1 in the definition of a graded back-and-forth system. Furthermore, the Lemma 3.2 shows that condition 2 from the definition of a graded back-and-forth system is satisfied. Therefore, the equivalence relations E_k form a graded back-and-forth system with an empty list of parameters.

For any finite list of parameters $\overline{c}$, we define the equivalence relations $E_k^{\overline{c}}$ by $\overline{x}E_k^{\overline{c}}\overline{y}$ if and only if $\overline{cx}E_k\overline{cy}$. By this definition, $\overline{x}E_0^{\overline{c}}\overline{y}$ if and only if the subgraph consisting of the points in $\overline{c}$ and $\overline{x}$ is isomorphic to the subgraph consisting of the points in $\overline{c}$ and $\overline{y}$. Therefore, for all atomic formulas $\psi(\overline{u})$ in the language of graph theory with constants for $\overline{c}$, $\psi(\overline{x})$ holds in $(G,\overline{c})$ if and only if $\psi(\overline{y})$ holds in $(G,\overline{c})$. Furthermore, suppose that $\overline{x}E_{k+1}^{\overline{c}}\overline{y}$ and a is any element of G. By definition of $E_{k+1}^{\overline{c}}$, $\overline{cx}E_{k+1}\overline{cy}$ and by Lemma 3.2, there is a b such that $\overline{cx}aE_k\overline{cy}b$. Applying the definition of $E_k^{\overline{c}}$ again, we have $\overline{x}aE_k^{\overline{c}}\overline{y}b$. Therefore, for any finite list of parameters $\overline{c}$, the relations $E_k^{\overline{c}}$ form a graded back-and-forth system for $(G,\overline{c})$. Furthermore, it is clear from the definition of $E_k^{\overline{c}}$ that $\overline{x}E^{\overline{c}}\overline{y}$ if and only if the neighborhood $N(\overline{cx},7^k)$ with constants for the parameters $\overline{c}$ and $\overline{x}$ is isomorphic to the neighborhood $N(\overline{cy},7^k)$ with constants for the parameters $\overline{c}$ and $\overline{y}$. Therefore, for any n and k, there are only finitely many equivalence classes of n-tuples from G under the relation $E_k^{\overline{c}}$ and each of these equivalence classes is describable by a Σ_2 formula in the language of graph theory with constants for $\overline{c}$. Let $\Gamma^{\overline{c}}$ denote the set of all formulas defining these equivalence classes.

Lemma 3.5. *Fix a (possibly empty) finite tuple $\overline{c}$ of parameters from G. Every formula $\theta(\overline{x})$ in the language of graph theory with constants from $\overline{c}$ is equivalent in G to a boolean combination of the formulas in $\Gamma^{\overline{c}}$.*

Proof. This follows immediately from Lemma 3.4 since we have already verified the hypotheses for our graded back-and-forth system $E_k^{\overline{c}}$, $k \in \omega$. $\square$

Theorem 3.6. *G is a computably categorical graph for which there is no Scott set of finitary formulas.*

Proof. First, we show that G is computable categorical by describing a computable isomorphism between arbitrary computable copies G_1 and G_2 of G. For $j \in \{1,2\}$, there is a one-to-one c.e. enumeration μ_j of $\mathcal{A}$ defined from G_j as follows. We identify the coding locations in G_j as the points x such that xRx and list them as c_i^j for $i \in \omega$. For each coding location c_i^j, we enumerate the corresponding set B_i^j by listing the cycles attached to c_i^j and dividing their lengths by 7. Because $\mathcal{A}$ has a unique one-to-one c.e. enumeration, there is a computable permutation f of ω such that $\mu_0(i) = \mu_1(f(i))$. We map c_i^0 to $c_{f(i)}^1$ and then match up the cycles attached to these coding locations. This map gives the desired computable isomorphism.

Second, we fix an arbitrary finite tuple $\overline{c}$ of parameters and show that G does not have a defining family of finitary formulas using these parameters. Since G is rigid, this implies that G has no Scott family of finitary formulas with these parameters. Let a be the coding location of a set A_i such that for infinitely many s, there is an $m \neq i$ for which $A_{i,s} \subseteq A_{m,s+1}$ and such that none of the parameters in $\overline{c}$ is in the same component as a. Suppose $\psi(\overline{c},x)$ is a formula which defines a. To get a contradiction, it suffices to find $b \neq a$ which also satisfies $\psi(\overline{c},x)$.

By Lemma 3.5, we can assume that ψ is a boolean combination of formulas from $\Gamma^{\overline{c}}$ and that it is written in disjunctive normal form as a combination such formulas. Let $\psi'(\overline{c},x)$ be the disjunct that is satisfied by a and write ψ' as

$$\theta_0(\overline{c},x) \wedge \theta_1(\overline{c},x) \wedge \cdots \wedge \theta_n(\overline{c},x) \wedge \neg\delta_0(\overline{c},x) \wedge \cdots \wedge \neg\delta_m(\overline{c},x)$$

where each of the formulas $\theta_i(\overline{c},x)$ and $\delta_i(\overline{c},x)$ describe a neighborhood of the form $N(\overline{c}x,7^k)$ for some $k \in \omega$. It suffices to find $b \neq a$ which satisfies $\psi'(\overline{c},x)$. In choosing b, we restrict our attention to coding locations for sets involved in infinitely many dumping operations and we assume that b does not lie in the same component with any of the parameters from $\overline{c}$.

Each formula $\theta_j(\overline{c}, x)$ describes a neighborhood $N(\overline{c}x, 7^{k_j})$ satisfied by a. In particular, θ_j can be described by two properties.

1. Since a is not in the same component with any of the parameters from $\overline{c}$, θ_j implies that $N(\overline{c}x, 7^{k_j})$ is the disjoint union of $N(\overline{c}, 7^{k_j})$ and $N(x, 7^{k_j})$.

2. Since a is the coding location for an infinite set A_i, θ_j says x is the coding location for a set X such that $X|k_j = A_i|k_j$ and such that X has at least k_j many elements greater than k_j.

Let k be the maximum of the sizes of neighborhoods described by $\theta_0, \ldots, \theta_n$. Any element b which does not lie in the same component with any of the parameters from $\overline{c}$ and which is the coding location for a set Y such that $Y|k = A_i|k$ and such that Y has at least k many elements greater than k will satisfy all of the $\theta_j(\overline{c}, x)$ formulas.

Each formula $\delta_j(\overline{c}, x)$ describes a neighborhood $N(\overline{c}x, 7^{l_j})$ which is not satisfied by a. For this neighborhood, δ_j could say either that x is not a coding location, or that $N(\overline{c}, 7^{l_j}) \cap N(x, 7^{l_j}) \neq \emptyset$ or that x is the coding location for a set X but that either $X|l_j \neq A_i|l_j$ or X does not have at least l_j many elements greater than l_j. Let l be the maximum of the l_j parameters and let K be the maximum of k and l. From this description, it is clear that if b does not lie in the same component with any of the parameters $\overline{c}$ and b is the coding location for a set Y such that $Y|K = A_i|K$ and Y has at least K many elements greater than K, then b will satisfy all θ_j and all $\neg\delta_j$ and hence will satisfy ψ'. Therefore, it suffices to find a $b \neq a$ which meets these conditions.

Let s be a stage in our construction of $\mathcal{A}$ such that $A_{i,s}|K = A_i|K$ and $A_{i,s}$ contains at least K many elements greater than K. By property (b) of condition 3 of Theorem 2.2, we can assume that s is such that for every $t > s$ and every $j \neq i$, if A_i is dumped into A_j at stage t (so $A_{i,t} \subseteq A_{j,t+1}$), then $A_{j,t}$ does not contain any numbers $\leq K$. Furthermore, by property (d) of condition 3 of Theorem 2.2, $A_{j,t+1} = A_j$ and hence $A_i|K = A_j|K$ and A_j contains at least K many elements greater than K. Finally, let z be such that every component of G containing a parameter from $\overline{c}$ codes a set from $\mathcal{A}$ with index less than z. By property (c) of condition 3 of Theorem 2.2, we can assume that s is such that if A_i is dumped into A_j after s, then $j > z$ and hence the component of G corresponding to A_j does not contain a parameter from $\overline{c}$. Since A_i is dumped into infinitely many sets, let $j \neq i$ be an index such that for some $t > s$, A_i is dumped into A_j at stage t. By the comments in this paragraph, A_j is a set such that $A_j|K = A_i|K$, A_j contains at least K many elements greater than K and the component of G corresponding to A_j does not contain any of the parameters from $\overline{c}$. Let b be the coding location for the set A_j. By the arguments of the previous paragraph, b satisfies ψ' and $b \neq a$, as required. $\square$

References

[1] C.J. Ash, J.F. Knight, M. Manasse & T. Slaman, *Generic copies of countable structures*, Annals of Pure and Applied Logic vol. 42 (1989), 195-205.

[2] J. Chisholm, *Effective model theory versus recursive model theory*, Journal of Symbolic Logic vol. 55 (1990), 1168-1191.

[3] P. Cholak, S. Goncharov, B. Khoussainov & R.A. Shore, *Computably categorical structures and expansions by constants*, Journal of Symbolic Logic vol. 64 (1999), 13-37.

[4] S.S. Goncharov, *Autostability and computable families of constructivizations*, Algebra and Logic vol.14 (1975), 392-409 (English translation).

[5] S.S. Goncharov, *The quantity of non-autoequivalent constructivizations*, Algebra and Logic vol.16 (1977), 169-185 (English translation).

[6] D.R.Hirschfeldt, *Degree spectra of intrinsically c.e. relations*, Journal of Symbolic Logic vol.66 (2001), 441-469.

[7] D.R. Hirschfeldt, B. Khoussainov, R.A. Shore, *A computably categorical structure whose expansion by a constant has infinite computable dimension*, Journal of Symbolic Logic vol. 68 (2003), 1199-1241.

[8] W.Hodges, *Model theory*, Cambridge University Press, Cambridge, 1993.

[9] O.V. Kudinov, *An autostable 1-decidable model without a computable Scott family of $\exists$ formulas*, Algebra and Logic vol. 35 (1996), 255-260 (English translation).

[10] D.Marker, *Model Theory: An Introduction*, Springer–Verlag, Heidelberg, 2002.

[11] V.L. Selivanov, *Enumerations of families of general recursive functions*, Algebra and Logic vol.15 (1976), 128-141 (English translation).

[12] R.I. Soare, *Recursively enumerable sets and degrees*, Springer–Verlag, Heidelberg, 1987.