Annals of Pure and Applied Logic 166 (2015) 851–880

Contents lists available at ScienceDirect

Annals of Pure and Applied Logic

www.elsevier.com/locate/apal

On Δ^0_2-categoricity of equivalence relations [☆]

Rod Downey, Alexander G. Melnikov, Keng Meng Ng [*]

ARTICLE INFO

Article history:
Received 4 August 2014
Received in revised form 31 March 2015
Accepted 3 April 2015
Available online 5 May 2015

MSC:
03D45
03C57
03D25

Keywords:
Equivalence structures
Computable mathematics
Categoricity
Isomorphisms

ABSTRACT

We investigate which computable equivalence structures are isomorphic relative to the Halting problem.

1. Introduction

This paper is within the scope of two frameworks: the first one investigates effective properties of equivalence relations (to be discussed in Section 1.1), and the other one studies non-computable isomorphisms between computable structures (see Section 1.2). The main idea of this paper can be described as follows: We view a computable equivalence structure as an abstraction to the situation when a computable algebraic structure has several *components*. Examples of such structures include direct or cardinal sums of groups or rings, shuffle and free sums of Boolean algebras, and graphs having several connected components. We simplify the situation by essentially removing all algebraic content from each component, so that we have to care only about matching the sizes of components correctly when we construct an isomorphism. The idea is that to understand the general situation, we should *first* understand the much simpler associated setting where the algebra has been stripped away. In particular, the present paper is a companion to Downey, Mel-

[☆] The first and the second author were partially supported by Marsden Fund of New Zealand. The third author is partially supported by the MOE grants MOE2011-T2-1-071 and MOE-RG26/13. The authors would like to thank the anonymous referee for helpful comments.

[*] Corresponding author.
E-mail address: kmng@ntu.edu.sg (K.M. Ng).

http://dx.doi.org/10.1016/j.apal.2015.04.003
0168-0072/© 2015 Elsevier B.V. All rights reserved.

nikov and Ng [15], where p-groups are associated with equivalence relations. One might expect that it would be easy to understand Δ^0_2-isomorphisms for these "degenerate" computable structures, particularly ones as simple as equivalence relations. We will see that the subject is a lot deeper than one might expect. Having abstracted the algebraic properties into a setting with no apparent *algebraic* difficulties, now one faces many *computability-theoretic* difficulties in such studies. Indeed, we see that a non-standard $\emptyset'''$-technique is required to answer a very basic (but fundamental) question. The proof is of some purely technical interest; its high complexity also partially explains why so little is known about Δ^0_2-isomorphisms between computable structures in general. We now turn to a more detailed discussion and background.

1.1. Effectively presentable equivalence structures

Arguably, the study of effective reducibilities between countable equivalence relations goes back to Mal'cev who founded the theory of numberings (see Ershov [16] for a detailed exposition). Many results of numbering theory can be translated into results on equivalence relations and vice versa, see the recent paper [1] for more details. Numbering theory has been one of the central topics in the Soviet logic school for over 40 years. In the West, the topic has traditionally received less attention (but see Lachlan [25]), and it is fair to say that it did not occupy center stage in computability theory.

Recently however, the subject has enjoyed a rapid development, partially because of the simultaneous and successful development of the theory of Borel equivalence relations, see textbook [6]. The theory of effective equivalence relations has grown to a rather broad area; we cite [1,18,9] for recent results on this subject. Many results of this paper can be stated in terms of Δ^0_2-embeddings between effectively presented equivalence structures. However, we choose a different approach (see the next subsection) and thus we will not provide any further background on effective reducibilities between equivalence structures.

1.2. Non-computable isomorphisms between computable structures

Recall that a structure is *computable* if its open diagram is a computable set. Recall that a computable algebraic structure $\mathcal{A}$ is *computably categorical* if any computable structure $\mathcal{B}$ isomorphic to $\mathcal{A}$ is *computably* isomorphic to $\mathcal{A}$. In many common classes computable categoricity can be understood as a synonym of being algebraically tame. For example, it is well-known that a computable linear order is computably categorical iff it has finitely many adjacencies, a computable Boolean algebra is computably categorical off it has finitely many atoms, and there is a full and simple description of computably categorical abelian p-groups, see [27,31,32] and [3,17] for further examples. Most of these classes are not effectively universal (i.e., these structures cannot effectively encode an arbitrary computable arbitrary graph, see [11]). On the other hand, when we want to study more complex algebraic structures and their computability theory, we often need to abandon computable categoricity. As a consequence, there has been an increasing interest in *non-computable* isomorphisms between computable structures.

The central notion in the study of non-computable isomorphisms is:

Definition 1.1. Let $n > 1$ be a natural number. A computable algebraic structure A is Δ^0_n-categorical if every two computable presentations of A are $0^{(n-1)}$-isomorphic.[1]

In contrast to computable categoricity, obtaining a complete classification of Δ^0_n-categoricity in a given class is typically a difficult task. Already for $n = 2$ and even for algebraically very well understood classes,

[1] Here $0^{(n+1)}$ stands for the $(n+1)$th iterate of the Halting problem. We note that there are variations of Definition 1.1 such as the notion of relative Δ^0_n-categoricity [3], and also related notions of categoricity spectra [20] and degrees of categoricity [19,10].

the problem may be challenging. The study of Δ^0_n-categorical structures has some independent technical interest as such investigations typically require new ideas and techniques (see, e.g., [2,12,13]). As a consequence of these technical difficulties, our knowledge of Δ^0_n-categorical structures is rather limited even when $n = 2$. Only recently, there has been significant progress in understanding Δ^0_n-categoricity in several specific classes, for small n. It follows from [8,30] that every free (non-abelian) group of rank ω is Δ^0_3-categorical, and the result cannot be improved to Δ^0_2. It is also known that every computable completely decomposable group is Δ^0_5-categorical, and the result is sharp [13]. Every computable homogeneous completely decomposable group is Δ^0_3-categorical, and a group of this form is Δ^0_2-categorical if and only if it encodes a semi-low set into its divisibility relation [12]. See also [29,28,4,5] for more results on Δ^0_n-categorical structures for small n. We emphasize that most of the results discussed above are technically quite difficult, and some of these results require new algebraic or computability theoretic techniques, and sometimes both.

1.3. Results

As we noted above, the study of Δ^0_2-categorical structures already tends to be technically difficult both algebraically and computability-theoretically. We would like to pick a very tame algebraic class where we could concentrate only on the *computability-theoretic* aspects of Δ^0_2-isomorphisms. The class of computable equivalence classes is as algebraically simple as it could get, yet we have good evidence that equivalence structures have interesting *effective properties* as we discussed in Section 1.1 above.

In the context of this paper, a computable equivalence structure is an abstraction to the situation when a computable algebra has several *components*; e.g., think of a cardinal or direct sum of algebras, or imagine a graph with several connected components. We remove all structure from each component and keep only one fundamental property:

From stage to stage, a component can only increase in size.

To build a Δ^0_2-isomorphism, we need to (at least) match the sizes correctly. In a companion paper [15] we consider abelian groups of Ulm type 1. This is a slightly more complicated class. It reflects the situation when “components” are *not* invariant under automorphisms and have to be *guessed.*

Calvert, Cenzer, Harizanov and Morozov [7] observed that every computable equivalence structure is Δ^0_3-categorical, and they gave several sufficient conditions for a computable equivalence structure to be Δ^0_2-categorical. We address the following problem left open in [7]:

Which computable equivalence structures are Δ^0_2-categorical?

A computable equivalence structure E is uniquely described, up to isomorphism, by the Σ^0_2 *multiset* of sizes of classes that occur in the structure. Thus, the question above is really a question about Σ^0_2-multisets. Our understanding of Σ^0_2-multisets is very limited, and we wish to reduce the the question to the more familiar case of Σ^0_2-sets. Given an equivalence structure E, keep only one equivalence class for each finite size (i.e., remove repetitions of sizes). Call the resulting equivalence structure $\widehat{E}$ the *condensation* of E. If E is computably presentable, then so is $\widehat{E}$ (to be discussed in Section 2). We arrive at:

Is it true that E is Δ^0_2-categorical if and only if $\widehat{E}$ is?

The first main result of the paper answers the question in the affirmative:

Theorem 1.2. *Let E be a computable equivalence structure. Then E is Δ^0_2-categorical if and only if its condensation $\widehat{E}$ is Δ^0_2-categorical.*

Theorem 1.2 came to us as a surprise, as it seemed that having multiple classes of the same size was a natural viable property we can use to diagonalize. However, the hope for carrying out such a diagonalization had an unavoidable non-uniform blockage. This is exploited for the proof of Theorem 1.2. We remark that the proof has significant combinatorial complexity. Theorem 1.2 also reduces the main problem to the study of Σ^0_2-sets of a special kind, as we explain below.

From the Δ^0_2-categoricity point of view, the only non-trivial case is when an equivalence structure E has infinitely many infinite classes, and arbitrarily large finite classes. We shall call such equivalence structures *non-degenerate*. Given a set X, define $E(X)$ to be a non-degenerate equivalence structure having exactly one class of size x for every $x \in X$. We emphasize that X is Σ^0_2 iff $E(X)$ has a computable copy (folklore). For notational convenience, we omit "Δ^0_2" when we speak about sets:

Definition 1.3. We say that an infinite Σ^0_2-set X is *categorical* if $E(X)$ is Δ^0_2-categorical.

Theorem 1.2 reduces the Δ^0_2-categoricity problem to the question:

Which Σ^0_2 sets are categorical?

No classical notion of computability theory seems to capture categoricity of a Σ^0_2 set. We compare categoricty of a Σ^0_2-set to some other properties that occur in effective structure theory. It seems that the answer might lie in sets with weak guessing procedures for membership, such as *low* or *semilow* sets (soon to be described). In Theorem 4.2 we prove that every d-c.e. semi-low$_{1.5}$ set is not categorical, but the converse fails. Our interest in semi-low$_{1.5}$ sets is motivated by the recent results on Δ^0_2-categorical completely decomposable groups, where semi-lowness actually captures Δ^0_2-categoricity [12]. Semi-low and semi-low$_{1.5}$ sets play an important role in the theory of automorphisms of the lattice of c.e sets under set-theoretical operations [33].

We will see that each infinite limitwise monotonic set (to be defined in Section 2) is not categorical, but there exists a non-categorical Σ^0_2 set which is not limitwise monotonic (Theorem 4.1). Limitwise monotonic sets and functions naturally appear in the characterization of computable equivalence structures, direct sums of cyclic groups, and in many other contexts (see [23,24,14,22]). Limitwise monotonicity fails to describe categoricity of a Σ^0_2-set. Nonetheless, our intuition is that limitwise monotonicity "almost" captures (non-)categoricity of a Σ^0_2-set. Our second main result shows that the difference between non-categoricity and limitwise monotonicity is so subtle that c.e. degrees do not "see" this difference:

Theorem 1.4. *For a c.e. degree $\boldsymbol{a}$, the following are equivalent:*

(1) *$\boldsymbol{a}$ is high;*
(2) *$\boldsymbol{a}$ bounds an infinite set which is not limitwise monotonic;*
(3) *$\boldsymbol{a}$ bounds an infinite categorical set.*

We prove (2) ↔ (3) in Theorem 4.6 which will be stated in Section 4.2, and (1) ↔ (2) follows from [14]. To prove Theorem 4.6 we introduce a new computability-theoretic notion equal to the standard domination property [33] for high degrees. The new notion is much more convenient in the context of categorical sets; this new notion might be of some independent interest to the reader. We note that Theorem 1.4 continues the line of research into degrees bounding effective model-theoretic and algebraic properties, see survey [26].

2. Computable equivalence structures

Given a computable presentation of an equivalence structure E, we write $[i]$ for the equivalence class of the ith element in the representation, and we write $\#[i]$ for the size of $[i]$. We will denote the least element of the nth distinct equivalence class by c_n. That is, $c_0 = 0$ and c_{n+1} is the least number $i > c_n$ such that $[i] \neq [x]$ for any $x \leq c_n$. Let $C_n = [c_n]$. The sequence $\{C_n\}_{n\in\omega}$ is a uniformly c.e. sequence of pairwise disjoint sets. Without loss of generality, we may allow $C_n = \emptyset$ in case the equivalence structure has less than n classes. Conversely, given any uniformly c.e. sequence of pairwise disjoint sets $\{C_n\}_{n\in\omega}$, we can effectively and uniformly obtain a computable equivalence structure whose equivalence classes are exactly $\{C_n\}_{n\in\omega} - \{\emptyset\}$, and whose universe is $\cup_n C_n$. *Henceforth, we may think of a computable equivalence structure (relation) as of a uniformly c.e. sequence of pairwise disjoint sets.*

Definition 2.1. The characteristic of an equivalence relation E on ω is the set

$$\chi_E = \{\langle m, k\rangle : \ E \text{ has at least } k \text{ classes of size } m\},$$

where $k \in \omega$ and $m \in \omega \cup \{\omega\}$.

Evidently, $E \cong F$ if and only if $\chi_E = \chi_F$. We will also use $\chi_E^{fin} = \{\langle m, k\rangle \in \chi_E : \ \mathrm{m} \ \in \omega\}$ and $\pi_E = \{m : \langle m, 1\rangle \in \chi_E^{fin}\}$.

Recall that a total function $F : \omega \to \omega$ is limitwise monotonic [23,22,14] if there exists a total computable function $g(x, y)$ of two arguments such that $F(x) = \sup_y g(x, y)$ for every x. An infinite set is limitwise monotonic (l.m. for short) if it is the range of a limitwise monotonic function. It is well-known that a Σ_2^0 set is limitwise monotonic if and only if it contains an infinite limitwise monotonic subset [22]. Furthermore, an infinite limitwise monotonic set is always a range of some injective limitwise monotonic function [22].

Fact 2.2 *(Folklore).* An equivalence structure is computably presentable if and only if one of the following conditions holds:

(1) E has infinitely many infinite classes, and the set χ_E^{fin} is Σ_2^0, or
(2) E has finitely many infinite classes, the set χ_E^{fin} is Σ_2^0, and π_E is limitwise monotonic.

2.1. Categoricity of equivalence structures

This subsection contains the basic information about Δ_2^0-categoricity of equivalence structures. Recall that a computable structure $\mathcal{A}$ is relatively Δ_n^0-categorical if for each $\mathcal{B} \cong \mathcal{A}$ there is an isomorphism witnessing $\mathcal{B} \cong \mathcal{A}$ that is $\Delta_n^{0^{D_0(\mathcal{B})}}$, where $D_0(\mathcal{B})$ is the quantifier-free diagram of $\mathcal{B}$. Relative Δ_n^0-categoricity clearly implies Δ_n^0-categoricity.

Fact 2.3 *(Calvert, Cenzer, Harizanov, Morozov).* Every computable equivalence structure is relatively Δ_3^0-categorical. An equivalence structure is relatively Δ_2^0-categorical if and only if it has either finitely many infinite equivalence classes or π_E is finite.

Thus, only equivalence structures with infinitely many infinite classes and unbounded finite classes may be not Δ_2^0-categorical.

Definition 2.4. We say that a countable equivalence structure is *non-degenerate* if it has infinitely many infinite classes and the collection of sizes of its finite classes is an infinite set (i.e., arbitrarily large sizes occur).

We may accept the following:

Convention 2.5. From this point on, we will assume that all considered equivalence structures are non-degenerate.

Recall that $\#[i]$ stands for the size of class $[i]$. The proposition below will be used heavily.

Proposition 2.6. *For a computable (non-degenerate) equivalence structure E, the following are equivalent:*

(1) *E is Δ^0_2-categorical.*
(2) *In every computable copy of E, the size function $\# : \omega \to \omega \cup \{\infty\}$ is $\emptyset'$-computable.*

Proof. It is not difficult to check that (2) ⇒ (1). We prove (1) ⇒ (2). Recall that χ^{fin}_E is Σ^0_2, and there are infinitely many infinite classes by Convention 2.5. To see why (1) ⇒ (2), note that every computable equivalence structure E has a computable copy in which $\# \leq_T \emptyset'$. To produce such a copy, start with infinitely many infinite classes. Adjoin to these infinitely many infinite classes an equivalence structure defined by the following procedure. Set $\#[i]_s = \infty$ if χ^{fin}_E tells us that the class $[i]$ has to be changed. Then introduce a new class with the appropriate finite size representing $[i]$, and repeat. Note that we can always ask $\emptyset'$ if $\#[i] = \infty$. Now, if the structure is Δ^0_2-categorical, then we can use the Δ^0_2 isomorphism from the "regular" copy described above onto any copy to introduce the desired $\emptyset'$-procedure. □

Note that we could replace (2) of Proposition 2.6 above by (2′): *For every computable copy of E, there exists a $\emptyset'$-procedure for deciding if a given class is finite.*

Convention 2.7. The proof of Proposition 2.6 above shows that every computable equivalence structure has a presentation in which $\#$ is computable in $\emptyset'$. We will call this special copy *regular* or *standard*.

3. From multisets to sets

In this section we prove that repetitions of finite classes do not effect Δ^0_2-categoricity. We first prove a useful lemma which is interesting on its own right.

Lemma 3.1. *Suppose $X \subseteq Y$ are infinite Σ^0_2 sets. If Y is categorical then so is X.*

Proof. Given any computable presentation $\{C_n\}_{n\in\omega}$ of $E(X)$, we construct a computable copy $\{D_n\}_{n\in\omega}$ of $E(Y)$ and an isomorphic Δ^0_2-embedding g of $\{C_n\}_{n\in\omega}$ into $\{D_n\}_{n\in\omega}$. By our assumption, $E(Y)$ is Δ^0_2-categorical, and thus there is a Δ^0_2-function predicting the sizes of D_n correctly (see Proposition 2.6). We will use the isomorphic embedding g to define a Δ^0_2 size-function for $\{C_n\}_{n\in\omega}$. Proposition 2.6 and the arbitrary choice of $\{C_n\}_{n\in\omega}$ will imply that X is categorical.

We assume $\#C_{n,s} \neq \#C_{m,s}$ for every $n, m < s$, and that $C_{n,s} \neq \emptyset$ for all $n < s$. We also choose a Σ^0_2-approximation $(Y_s)_{s\in\omega}$ of Y so that at every stage s and every $n, m < s$ we have $\#C_{n,s} \in Y_s$. We build a computable equivalence structure $\{D_n\}_{n\in\omega}$, a total Δ^0_2 function g, and for every y, n meet

$$R_y : y \in Y \text{ if and only if } \exists! j\ \#D_j = y,$$

and

$$P_n : \exists s\, \forall t \geq s\, g(n)[t] \downarrow = g(n)[s] \text{ and } \#C_n = \#D_{g(n)[s]}.$$

Strategy for R_y. If $y \in Y_s$ and there is no $D_{k,s}$ of size y, then pick i fresh and define $D_{i,s}$ to be a class of size y. Say that $D_{i,s}$ is a witness for R_y. If R_y already has a witness $D_{i,s}$ and $y \notin Y_s$, then declare $\#D_{i,s} = \infty$ and say that R_y has no witness.

Strategy for P_n. If first initialized at stage s, define $g(n) = k$ such that $\#D_{k,s} = \#C_{n,s}$ (recall that $\#C_{n,s} \in Y_s$). Otherwise, if $g(n)$ was already previously defined, wait for $\#C_{n,s} > \#C_{n,s-1}$. Consider the cases:

Case 1. $\#C_{n,s+1} < n$. Suppose $\#C_{n,s+1} = \#D_{k,s+1}$ for some $D_{k,s}$ serving as a witness for R_y with $y < n$. Then declare $D_{g(n),s}$ infinite and reset $g(n)$ to be equal to k.

Case 2. $\#C_{n,s+1} \geq n$. Whenever $\#C_{n,s+1} = \#D_{k,s+1}$ for some $D_{k,s}$ serving as a witness for R_y with $y > n$, declare D_k infinite and initialize R_y by setting its witness undefined.

Construction. At stage 0, do nothing. At stage s, let R_y- and P_n-strategies with $y, n < s$ act according to their instructions.

Verification. Observe that at a stage s, if $y \in Y_s$ then either R_y has a witness or $\#D_{g(n),s} = y$ for some $n < s$. Go to a stage s so large that either $\#C_{n,s} > y$ or has already reached its finite limit. Then either $\#D_{g(n),t} = y$ for all $t > s$ or R_y has a stable witness. In both cases we have exactly one class of size y. We conclude that R_y is met. For P_n, go to a stage s so that either $\#C_{n,s} > n$ or C_n never changes after s. In both cases $g(n)$ is defined and will never be reset to a new value. We conclude that P_n is met as well. □

Recall that π_R stands for the set of sizes of finite classes that occur in R. Recall also that we restrict ourselves to the case when there are infinitely many infinite classes, and the sizes of finite classes are unbounded. Note that the set π_R is Σ^0_2 relative to R. Thus, if R is computable, then so is $E(\pi_R)$. For notational convenience, we denote $E(\pi_R)$ by $\widehat{R}$ and call it *the condensation of* R.

We next turn to an important question: If an equivalence structure R is Δ^0_2-categorical, must its condensation $\widehat{R}$ be Δ^0_2-categorical? What about conversely? In other words, does the repetition of finite equivalence classes affect Δ^0_2-categoricity? We prove that the answer is no. Thus we may restict the study of Δ^0_2-categoricity of equivalence structures to only those structures in which every finite class appears at most once.

The rest of this section is devoted to the proof of Theorem 3.2.

Theorem 3.2. *A computable equivalence structure R is Δ^0_2-categorical if and only if its condensation $\widehat{R}$ is Δ^0_2-categorical.*

First we swiftly dispose of the easy direction.

Lemma 3.3. *If a computable equivalence structure R is Δ^0_2-categorical, then its condensation $\widehat{R}$ is Δ^0_2-categorical as well.*

Proof. Consider the Σ^0_2 set $\{\langle m, k\text{-}1\rangle : k > 1, \langle m, k\rangle \in \chi^{fin}_R\}$. It corresponds to an equivalence structure having a computable copy V. Given a computable copy X of $\widehat{R}$, take a disjoint union Y of X and V. The resulting computable structure is a computable copy of R, and has a $\emptyset'$-computable function guessing sizes in Y correctly. Since the operation of taking the disjoint union is effective, we can restrict this function to the domain of X. □

We devote the rest of this section to the proof of the converse of Lemma 3.3. This direction turns out to be surprisingly combinatorially involved.

For the rest of this proof, we fix a computable listing $\{M_e\}_{e\in\omega}$ of all uniformly c.e. sequences. $M_e = \{M^e_i\}_{i\in\omega}$ is viewed as the eth equivalence structure (with possible repetition of finite classes). Given an equivalence structure M_e, we say that $\Phi^{\emptyset'}_j$ with range in $\{f, \infty\}$ is *a guessing function for* M_e if for every i, $\Phi^{\emptyset'}_j(i) = f$ iff $\#M^e_i < \infty$.

Lemma 3.4. *Predicate* $IND(i, e) \leftrightharpoons$ *"*$\Phi^{\emptyset'}_e$ *is a guessing function for* M_i*" is* Π^0_3.

Proof. $IND(i, e)$ holds if and only if

$$\Phi^{\emptyset'}_e \text{ is total and } \forall j \forall z \left(\Phi^{\emptyset'}_e(j) \downarrow= z \Rightarrow \left(z = f \Leftrightarrow \#M^i_j < \infty\right)\right)$$

This can be easily checked to be Π^0_3. □

It is not difficult to show that $IND(i, e)$ is Π^0_3-complete. Indeed, given any Π^0_3-predicate $\mathbb{P}$ and a pair (i, j), we can uniformly construct a guessing function $\Psi_{i,j}$ and a computable structure $E_{i,j}$ such that $\Psi_{i,j} = \#E_{i,j}$ iff $(i, j) \in \mathbb{P}$. We can make $\Psi_{i,j} = \infty$ on even and f on odd inputs. In $E_{i,j}$ we will have $\#[2k] = \infty$. We can make sure $\#[2k + 1]$ will be infinite for some k exactly if $\mathbb{P}$ fails on (i, j). Based on this observation, we conjecture that our complicated guessing procedure is necessary for the proof that will follow.

For the rest of this proof we now fix a computable equivalence structure $R = \{R_i\}_{i\in\omega}$ where finite classes may be repeated any number of times. We assume that the condensation $\widehat{R}$ is Δ^0_2-categorical. We fix a computable enumeration of the classes $\{R_i[s]\}_{i\in\omega}$ and assume that at every stage s there is at most one $i < s$ such that $R_i[s] \neq R_i[s - 1]$ and in the case where i exists we have $\#R_i[s] = \#R_i[s - 1] + 1$. Our goal is to produce (not uniformly in an index for R) a guessing function for R.

During the construction we build a computable structure $M = \{M_i\}$ and appeal to the Recursion Theorem to give us an index for M in advance (say index c).

3.1. Tree of strategies

Our tree of strategies is a version of the Baire space where the outcomes of each node is labeled $0 < 1 < \cdots$. Each node on the lth level is devoted to measuring if $IND(c, l)$ holds. Since this predicate is Π^0_3 there is an obvious way to computably approximate this using the outcomes of each node σ. We let $\{V^l_k\}_{l,k\in\omega}$ be a computable collection of c.e. sets such that $IND(c, l)$ holds iff $\#V^l_k < \infty$ for every k. Thus we naturally associate each outcome k of σ with the Σ^0_3 outcome where $\#V^l_k = \infty$. The Π^0_3 outcome of σ where $IND(c, |\sigma|)$ holds corresponds to the situation where every outcome of σ is visited finitely often. Since this latter outcome is a global outcome we will not need to place a corresponding σ-outcome for it.

We have a global commitment to make M a copy of the condensation (by Lemma 3.1 it is sufficient to make M a structure on a subset of the condensation). Since we are given an index for M in advance we know that there will be a *true node* σ_{true} of the construction. Namely σ_{true} is the leftmost node visited infinitely often such that each $\sigma * i$ is visited finitely often (the true node will later be formally defined). We will then use $\Phi^{\emptyset'}_{|\sigma|}$ to help build a guessing function for R. Since we have to guess at the true node we have to allow each node on the priority tree to have its own opinion about how the guessing function for R is to be defined. This will be maintained via *cliques* and *links*.

3.2. Cliques and links

A clique $\mathcal{C}$ is a collection of at least one class (possibly more) of R and will always have an associated link $\ell(\mathcal{C})$. This link points to a single M class. Intuitively every R-class of a clique is collectively associated with the M-class $M_{\ell(\mathcal{C})}$. A link for a clique will be fixed (will never be reassigned to another member of M)

until the clique is removed. A clique may grow when more R-classes join the clique but will never reduce in members. When a clique is removed the associated link is also removed.

Sometimes an R-class R_i which is not currently in a clique will also be linked to a class in M. We denote this link as $\ell(i)$. Again this means that R_i is associated with the M-class $M_{\ell(i)}$. Like a link for a clique, this link (for the class R_i) will be fixed until it is removed. This link will be removed if the class R_i joins a clique, or if a higher priority node acts. An R-class R_i for which $\ell(i)$ is defined is simply said to be linked. The intuitive idea is that a link denotes that we believe an R-class is finite and hence the linked element in M should also be the same. A clique denotes that we believe a collection of R-classes will all be infinite and that the linked element in M will also grow to infinity.

Each node σ on the tree of strategies will have its own separate version of cliques and links. For this reason we will often use the term σ cliques and σ links. If $\mathcal{C}$ is a clique we write $\min \mathcal{C}$ to be the index of the smallest member of $\mathcal{C}$, i.e. $\min \mathcal{C} = \min\{i \mid R_i \in \mathcal{C}\}$. We write $\operatorname{size}\mathcal{C}[s] = \min\{\#R_m[s] \mid m \in \mathcal{C}\}$, i.e. the size of the smallest class in $\mathcal{C}$.

3.3. Description of the proof

Since the proof of Theorem 3.2 is somewhat combinatorially involved, we will describe the main ideas behind the proof here. Most of the steps in the construction and verification are technical and are included simply to make the combinatorics work. Nevertheless there are several key ideas which will form the skeleton of the proof.

3.3.1. The simple case

We first assume the simple case when the condensation $\widehat{R}$ of R is effectively Δ^0_2-categorical [15, Definition 1.2]. This means there exists a computable procedure which, given an index of a computable copy of $\widehat{R}$, returns a Δ^0_2-index for # in that copy. It is not hard to show [15, Theorem 2.5] that R is effectively Δ^0_2-categorical. We sketch a different proof here, along the lines of the proof of Theorem 3.2.

We shall build a computable presentation $M = \{M_n\}_{n\in\omega}$ of $\widehat{R}$. Since $\widehat{R}$ is effectively categorical, by applying the Recursion Theorem, we have during the construction of M a computable approximation $g(n, s)$ where for every n, $\lim_s g(n, s)$ exists and equals f iff $\#M_n < \infty$ and equal ∞ iff $\#M_n = \infty$.

The basic plan is straightforward. We monitor each class R_i and associate with it some class M_n. To help organize this we declare R_i to be “linked” to class M_n, and we write $\ell(i) = n$. Naively we want to keep $\#M_{\ell(i)} = \#R_i$ and use $g(\ell(i))$ to predict $\#R_i$. Unfortunately we may have $\#R_i = \#R_j$ for some $i \neq j$ but we are committed to making M a structure on the condensation $\widehat{R}$. Thus we have to redirect at least one of the two links $\ell(i)$, $\ell(j)$ when we find that $\#R_i = \#R_j$. We want to ensure that each link $\ell(i)$ is redirected only finitely often. If this can be done then it is $\emptyset'$-computable to figure out the final stable object for each R_i and to read off $\lim_s g(\ell(i), s)$.

Hence during the construction when we see $\#R_i = \#R_j$, for $i < j$ we will immediately grow $M_{\ell(j)}$ to infinity, dissolve the link $\ell(j)$ and set up a new link $\ell(j) = \ell(i)$. What can happen next is that one of the two classes R_i, R_j grows. Suppose $\#R_i > \#R_j = \#M_{\ell(i)}$. In this case it is no good to keep $\#M_{\ell(i)} = \#R_j$ because otherwise the link $\ell(i)$ will point at a potentially finite class $M_{\ell(i)}$ even though $\#R_i$ can be ∞, and so the link $\ell(i)$ is of no use to us in deciding $\#R_i$. Therefore we should grow $\#M_{\ell(i)}$ to match $\#R_i$ whenever R_i grows, which means that $\ell(j)$ should be reassigned elsewhere because it is now pointing at the class $M_{\ell(i)}$ where $\#M_{\ell(i)} > \#R_j$, and thus $\#M_{\ell(j)} = \#M_{\ell(i)} > \#R_j$ will again tell us nothing about $\#R_j$. So if R_i grows before R_j we will be forced to reassigned either $\ell(i)$ or $\ell(j)$. If instead the class R_j grows first before R_i, we face a similar dilemma.

For general priority reasons, to resolve this situation, we should choose to keep $\ell(i)$, reassign $\ell(j)$ and grow $\#M_{\ell(i)} = \#R_i$. The issue now is that R_j may clash infinitely often with R_i, $i < j$ this way, and each time $\ell(j)$ is sacrificed by being reassigned to a fresh M-class, and in the end there is no stable link on j.

This is bad because $\emptyset'$ is unable to determine if the class R_j is finite; even though $\#R_j$ is necessarily infinite if $\ell(j)$ is reassigned infinitely often, but this latter fact is not decidable using only a $\emptyset'$ oracle.

The reader should realise that we have not yet made use of the function g; this function must obviously be used in an essential way. The idea is to introduce two kinds of objects in the construction; a *link* $\ell(i)$ and a *clique* $\mathcal{C}$ with pointer $\ell(\mathcal{C})$. A link $\ell(i)$ is a pointer associated with a single class R_i, while a clique is a collection of classes $\{R_i : i \in \mathcal{C}\}$ which collectively point at the class $M_{\ell(\mathcal{C})}$ (see Section 3.2).

These two objects pursue essentially opposing strategies. If i is linked to $M_{\ell(i)}$ then we will keep $\#M_{\ell(i)} = \#R_i$; whenever R_i grows, we must grow $M_{\ell(i)}$ accordingly. On the other hand a clique $\mathcal{C}$ will keep $\#M_{\ell(\mathcal{C})} = \operatorname{size}\mathcal{C} = \min\{\#R_i \mid i \in \mathcal{C}\}$, i.e. the size of the smallest class in $\mathcal{C}$. The decision as to whether we should have a link or a clique on a class R_i is determined by $g(\ell(i), s)$.

More specifically, for each R_i, we initially start off with a link $\ell(i)$ on i. We keep $\#M_{\ell(i)} = \#R_i$. When we find $g(\ell(i), s) = \infty$, we form a clique $\mathcal{C} = \{i\}$ with pointer $\ell(\mathcal{C}) = \ell(i)$, and remove $\ell(i)$. (This is step (2.1) of the construction.) While $g(\ell(\mathcal{C}), s) = \infty$ we keep $\#M_{\ell(\mathcal{C})} = \operatorname{size}\mathcal{C}$ and grow the clique by adding $j > i$ to $\mathcal{C}$ whenever $\#R_j \geq \operatorname{size}\mathcal{C}$. (This is step (2.2) of the construction.) If ever we see that $g(\ell(\mathcal{C}), s)$ changes its mind and takes value f, we will dissolve the clique $\mathcal{C}$ by removing $\ell(\mathcal{C})$ and restoring the link $\ell(i) = \ell(\mathcal{C})$.

Since we must make M a structure on the condensation $\widehat{R}$, we have to sort out any conflict in sizes. For instance, when we find $\#M_{\ell_0} = \#M_{\ell_1}$ where ℓ_0 and ℓ_1 are objects pointing at different M-classes, we will retain the M-class associated with an object of the highest priority (say M_{ℓ_0}) and declare $\#M_{\ell_1} = \infty$, and reassign ℓ_1 to now point at M_{ℓ_0}. (This is Phase 4 of the construction.) Priority amongst objects is determined by the value of the indices, i.e. the priority of $\ell(i)$ is i while the priority of $\ell(\mathcal{C})$ is $\min\mathcal{C}$, see "$\sigma = \tau$" under Section 3.3.3.

We now see that the problem described above is solved, and each i will eventually be involved in a stable link or clique. Suppose a class R_j has an object removed infinitely often because it conflicts with some R_i, for some least $i < j$. Assume that R_i already has a stable object ℓ. Now if $\#M_\ell < \infty$ then R_j cannot clash with R_i infinitely often, because eventually $\#M_{\ell(j)} > \#M_\ell$. Hence we must have $\#M_\ell = \infty$. Since M is always a substructure of the condensation $\widehat{R}$, we know that $\lim_s g(\ell, s)$ has to be correct, hence $g(\ell, -)$ will eventually take on the stable value ∞, we see that both i and j will eventually be involved in a clique $\mathcal{C}$. If R_i and R_j are both members of the same clique $\mathcal{C}$ then they both point at the same M-class and there are no further interactions between these two classes. So each R_i is eventually involved in a stable link or clique.

We now show that $\#R_i < \infty$ iff $\#M_\ell < \infty$ where ℓ is the stable object involving i: Since we always have $\#M_\ell \leq \#R_i$ (equality must hold when ℓ is a link), hence $\#R_i < \infty$ implies that $\#M_\ell < \infty$. On the other hand suppose that $\#R_i = \infty$. Then the stable object ℓ cannot be a link because otherwise M_ℓ is always grown to match the size of R_i, and so $\#M_\ell = \infty$ and we must have $\lim_s g(\ell, s) = \infty$, which in turn means that a clique containing i will be formed eventually. Thus the stable object ℓ must instead be a clique, and so we have $\#M_\ell = \infty$ (else $\#M_\ell < \infty$ and so $\lim_s g(\ell, s) = f$ and we would eventually dissolve the clique $\mathcal{C}$). Since the strategy for a clique always maintains $\#M_\ell = \operatorname{size}\mathcal{C}$, we see that $\#R_j = \infty$ for *every* member $j \in \mathcal{C}$.

Now to figure out if each class R_i is finite we may use $\emptyset'$ to first search for a stable object ℓ involving i, and then computing $\lim_s g(\ell, s)$. Since $\lim_s g$ is never wrong, its value will decide $\#M_\ell$ and hence $\#R_i$.

3.3.2. Introducing injury

We now describe the problems caused by considering "injury" in the formal construction. We illustrate this in a simplified setting. We now relax the condition that "$\widehat{R}$ is effectively Δ^0_2-categorical" to one that assumes that the procedure which returns a Δ^0_2-index for $\#$ in a computable copy of $\widehat{R}$ is $\emptyset''$-computable (instead of computable in the previous discussion). That is, there is a function $F \leq_T \emptyset''$ such that for every e,

$\Phi^{\emptyset'}_{F(e)}$ gives the size function # in the computable copy M of $\widehat{R}$ if M has index e. (Note that $F \leq_T \emptyset'$ is equivalent to being effectively Δ^0_2-categorical.)

It is easy to see that the condition $F \leq_T \emptyset''$ is equivalent to the existence of a computable function H such that for every e, given a computable copy $M = \{M_x\}_{x\in\omega}$ of $\widehat{R}$ with index e, the function $\Phi^{\emptyset'}_{H(e)}(x)$ is equal to the size function # of M on almost every class M_x. Under this assumption we describe how to prove that R is Δ^0_2-categorical.

We use the same setup as before. In this case we may have that $\lim_s g(x,s)$ is incorrect for finitely many x. Suppose x_0 is such that $\#M_{x_0} = \infty$ but $\lim_s g(x_0,s) = f$, and R_{i_0} has a stable link $\ell(i_0)$ pointing at M_{x_0}. Carrying out the strategy above, we see that it is now possible for there to be infinitely many $j > i_0$ such that R_j gets an object reassigned infinitely often (due to conflicts with R_{i_0}). In fact, since $g(x_0,-)$ is eventually stable with value f, no clique will be formed to point at M_{x_0}. This is bad because there are now infinitely many classes R_j with no stable link or clique (even though $\lim_s g(x,s)$ is wrong on only finitely many x), and thus our argument above does not directly apply to show that R is Δ^0_2-categorical.

This construction in fact *does work* with a slight modification. A more ingenious argument must be applied to show that R is Δ^0_2-categorical. Notice that for each such x_0, we should have $\#M_{x_0} = \infty \neq \lim_s g(x_0,s)$. This is because if $\#M_{x_0} = f \neq \lim_s g(x_0,s)$, then every class R_j is affected by M_{x_0} only finitely often, and so the incorrect prediction of g on such a class M_{x_0} has no long term effect on the stability of an R_j object. Hence if some class R_j enters the clique associated with M_{x_0} and if R_j later grows larger than $\#M_{x_0}$ we can reassign R_j to point to a different M-class. For each j the link on R_j is redirected by M_{x_0} only finitely often.

Now for each x_0 such that $\#M_{x_0} = \infty \neq \lim_s g(x_0,s)$ assume that i_0 is the least such that R_{i_0} has a stable link pointing at M_{x_0}; hence $\#R_{i_0} = \infty$ as well. For each s, define the c.e. set $\Xi(s)$ to contain all indices $j > i_0$ such that $\#R_j[t] \geq \#R_{i_0}[t]$ for some $t > s$. (We refer the reader to Definition 3.5 for the formal definition of Ξ; the actual definition is somewhat more complicated due to various technicalities, but is similar in spirit to the one given here.)

Now if it is the case that for every s there is some $j_s \in \Xi(s)$ such that $\#R_{j_s} < \infty$ then we could define a limitwise monotonic function f by letting $f(x)$ follow the size of the smallest class currently in $\Xi(x)$. In that case it is easy to check that the range of f gives an infinite limitwise monotonic subset of the finite sizes of $\widehat{R}$, which is impossible because we assumed that $\widehat{R}$ is Δ^0_2-categorical by applying Theorem 4.1(i) and Lemma 3.1.[2] (Theorem 4.1(i) has an elementary and self-contained proof. Although it appears later in the paper, it does not introduce any circularity to our exposition.)

Thus it must be the case that there exists some s so that for every $j \in \Xi(s)$, $\#R_j = \infty$. This means that if R_j conflicts with R_{i_0} infinitely often then $j \in \Xi(s)$ and thus we can also conclude that $\#R_j = \infty$. Since there are only finitely many different i_0 and x_0, we can fix non-uniformly an s larger than all the associated values for all the i_0, x_0 (we call these classes "*finite junk*"). We can then argue that for almost every i, either R_i is involved in a stable link or clique, or else i is injured infinitely often by finite junk in which case i is a member of $\Xi(s)$. (See Lemma 3.22.) In this way $\emptyset'$ can decide the ultimate fate of each R_i.

3.3.3. Priority ordering

The nodes on the strategy tree are ordered lexicographically from left to right. If σ is to the left of τ then we may think of σ as having higher priority than τ. If σ and τ are comparable then we do not formally order σ and τ; the interactions between σ and τ are more intricate in this case.

We will instead define a priority ordering among links and cliques. This will be the key driving force of the construction and is used to regulate when cliques and links are formed and when they are allowed to get destroyed. The cases to consider are the following:

[2] In this discussion of the basic case we will in fact need to apply the uniform version of Lemma 3.1. That is, the index witnessing the categoricity of X can be obtained effectively in a Σ^0_2 index for Y and an index witnessing the categoricity of Y.

- *σ is to the left of τ*: In this case every σ link and σ clique is declared to be of higher priority than every τ link and τ clique.
- $\sigma = \tau$: A σ link $\ell_\sigma(i)$ is of higher priority than another $\ell_\sigma(i')$ iff $i < i'$. In the construction we will ensure that $i = i'$, i.e. we never have two different σ links on the same class simultaneously existing. A σ clique $\mathcal{C}$ is of higher priority than another σ clique $\mathcal{C}'$ iff $\min \mathcal{C} < \min \mathcal{C}'$; again in the construction we ensure that we never have two σ cliques with the same min simultaneously in existence. In fact, σ cliques are always pairwise disjoint. Finally a σ link $\ell_\sigma(i)$ is of higher priority than a σ clique $\mathcal{C}$ iff $i < \min \mathcal{C}$. In the construction we ensure $i \neq \min \mathcal{C}$; in fact $i \notin \mathcal{C}$.
- $\sigma \supset \tau$: Let k be such that $\sigma \supseteq \tau * k$. Every τ clique $\mathcal{C}$ and every τ link $\ell_\tau(i)$ with $\min \mathcal{C} < \min I_k^\tau$ (or $i < \min I_k^\tau$) is of higher priority than every σ clique and every σ link. Every τ clique $\mathcal{C}$ and every τ link $\ell_\tau(i)$ with $\min \mathcal{C} \geq \min I_k^\tau$ (or $i \geq \min I_k^\tau$) is of lower priority than every σ clique and every σ link. (The notation I_k^τ will be defined in Section 3.3.4, intuitively I_k^τ is the interval of influence of the τ-strategy.)

  For instance if $i < j$ are both in I_k^τ then $\ell_\tau(i)$ is of *lower* priority than $\ell_\sigma(j)$, even though the former is a link on a class with a smaller index.

It is a straightforward but somewhat tedious exercise to check that this gives rise to a linear ordering of all links and cliques in existence at any instance during the construction.

Intuitively the priority ordering is best described by the following. If σ is to the left of τ then each σ object is of higher priority than each τ object. If $\sigma * k \subseteq \tau$ then the priority of a σ object $\ell_\sigma(i)$ or $\ell(\mathcal{C})$ with $i = \min \mathcal{C}$ (we call this a (σ, i) object) depends on which interval I_m^σ the class i is in. Every (σ, i) object for $i \in \cup_{m<k} I_m^\sigma$ is of higher priority than every τ object. Every (σ, i) object for $i \in \cup_{m \geq k} I_m^\sigma$ is of lower priority than every τ object. In other words we allow τ objects to have higher priority over certain σ objects, even though $\sigma \subset \tau$.

3.3.4. Notations

We let $g_l(x, s)$ be a computable sequence of total functions with range in $\{\infty, f\}$ so that $\lim_s g_l(x, s) = \Phi_l^{\emptyset'}(x)$ if the latter converges, and where $\lim_s g_l(x, s)$ does not exist otherwise. Given a node σ and a stage s where σ is visited, we write $g_\sigma(x)[s]$ to mean $g_{|\sigma|}(x, s')$ where s' is the number of times where σ has been visited up to stage s. That is, we only update the approximation to $g_{|\sigma|}(x, -)$ whenever σ is visited.

Each node σ of the construction is associated with a finite sequence of finite intervals $I_0^\sigma, I_1^\sigma, \cdots$ of ω. Intuitively, the interval I_k^σ grows when outcome $\sigma * k$ is visited. We always have $\max I_k^\sigma + 1 = \min I_{k+1}^\sigma$ and $I_k^{\sigma * j} \subset I_j^\sigma$. The true node σ_{true} will be the only node to have every interval $I_k^{\sigma_{true}}$ stable (i.e., its definition will never be changed at a later stage), finite and non-empty.

To *initialize* a node σ means to remove all σ cliques and remove all σ links, and set $I_k^\sigma = \emptyset$ for every k.

The following definition keeps track of the effect of the “finite junk” arising in the construction. It will be used during the construction. Lemma 3.14 will make it clear why Definition 3.5 is necessary.

Definition 3.5. Let i be an index and s be a stage. Define the c.e. set $\Xi(i, s)$ by specifying the following computable enumeration of $\Xi(i, s)$. Let $\Xi(i, s)[t] = \{i\}$ for every $t \leq s$. At stage $t + 1 > s$ enumerate j into $\Xi(i, s)$ if $j \notin \Xi(i, s)[t]$ and one of the following holds:

- $\#R_j[t + 1] \geq \#R_k[t + 1]$ for some k such that $\#R_k[t + 1] > j$ where k is already in $\Xi(i, s)$, or
- $\#R_j[t + 1] \geq \#R_k[t + 1]$ for some k which was previously enumerated in $\Xi(i, s)$ at stage $t' \leq t$ and $\#R_k[t'] < \#R_k[t + 1]$.

In other words we enumerate j in $\Xi(i, s)$ at a stage $t + 1$ if the size of R_j currently exceeds (or is equal to) the size of another class R_k which was previously enumerated in $\Xi(i, s)$ but where the size of R_k has since grown. If $\#R_k[t + 1] > j$ then we can ignore the growth restriction on $\#R_k$.

As is customary in a priority construction, we use stage s not only to refer to a particular stage of the construction, but also to refer to a particular instance or a particular step of the construction within stage s. Some authors prefer to use the distinct term "sub-stage" instead.

3.3.5. Putting the construction on a tree

Finally we consider the general case when $\widehat{R}$ is Δ^0_2-categorical. Now guessing for the index of the size function of M is Π^0_3, so we carry out the construction on the priority tree defined in Section 3.1. Roughly speaking each node σ is given an interval $\cup_n I^\sigma_n$ to work in, and will carry out its own version of the basic strategy within its assigned interval. At the true node σ_{true}, the guessing function $g_{\sigma_{true}}$ is equal to the size function $\#$ of M. Each successor of σ_{true} is visited finitely often.

Now we need to distinguish between the parameters of different nodes. Hence, instead of links and cliques we shall have σ links and σ cliques. The priority ordering between different objects was defined in Section 3.3.3.

Now we fix $\sigma = \sigma_{true}$ and let $i_0 = \min I^\sigma_0$. That is, the true node σ is assigned the interval $[i_0, \infty)$ to work in. Let's try and briefly describe why for each $i \geq i_0$ we have that either $i \in \widehat{\Xi}$ or i is eventually involved in a stable σ link or a stable σ clique. Here we do not wish to encumber the reader with the precise definition of $\widehat{\Xi}$ (we refer the reader to Definition 3.16); it suffices at this point to say that $\widehat{\Xi}$ is more or less the union of $\Xi(s)$ for all infinite classes R_i, $i < i_0$ for a large enough s.

A key difference between this and the previously discussed cases is that due to the construction being carried out on a tree, the true strategy working for σ will not be able to act at every stage, only at infinitely many stages. This means that we have to ensure that at stages where σ is not active, the construction still respects the needs of every σ object. For instance if σ is active and finds $\#M_\ell > \#R_i$ then M_ℓ is no longer helpful in deciding $\#R_i$. Hence we should ensure that whenever there is a (σ, i) object ℓ we must at every stage keep $\#M_\ell \leq \#R_i$, unless a higher priority object demands otherwise, in which case ℓ should be removed. Note that in the case $\ell = \ell_\sigma(i)$ is a link then the strategy σ only needs to grow $\#M_\ell$ to be equal to $\#R_i$ whenever σ is visited; at non-σ-stages it is only important to keep $\#M_\ell \leq \#R_i$ (and not necessarily equal).

Let's assume that i is never part of a stable σ link or a stable σ clique. We explain why i should be in $\widehat{\Xi}$. By examining the priority between objects, there are only finitely many pairs (τ, k) such that a (σ, i) object ℓ can be removed by a conflict with a (τ, k) object ℓ' (of higher priority). The key to this analysis is to fix a large stage s^* (how large s^* needs to be is explained carefully in the verification; for now we assume it is large enough so that all higher priority activities are stable). We consider two cases: when ℓ' is formed before s^* and when it is formed after s^*.

If ℓ' is formed after s^* then necessarily we should have $\tau \subseteq \sigma$ (as all other nodes are either stable or of lower priority). In this case if $\tau = \sigma$ then we use the induction hypothesis, and if $\tau \subset \sigma$ then we must have $k < i_0$ and so $i \in \widehat{\Xi}$. From σ's point of view (τ, k) belongs to the "finite junk" which σ must accept as a finite parameter given non-uniformly, so we can build (τ, k) into the definition of $\widehat{\Xi}$.

Now if ℓ' is formed before s^* (there are only finitely many such objects) then we will see that the target class $M_{\ell'}$ must eventually be infinite. In that case $g_\sigma(\ell')$ must eventually take on value ∞ and consequently ℓ must be a σ clique. In that case the strategy of ℓ strategy will switch to a negative strategy, and will never again request for $M_{\ell'}$ to increase. If ℓ' is a historical object associated with some τ which is never again active then there is no need for ℓ to be removed; neither ℓ nor ℓ' will request for $M_{\ell'}$ to be increased and so both objects can co-exist.

This forms the main ideas behind the machinery of the construction. The formal construction and verification will address the multiple technical complications which arise in the implementation of these ideas.

3.4. Construction

At stage 0 initialize every node and do nothing else. Suppose we are at stage $s > 0$. The construction splits into phases:

Phase 1, Defining δ_s and initialization.

(1.1) We define the stage s approximation δ_s to the true node. We will have $|\delta_s| = s$ and this is defined inductively as follows. If $\delta_s \upharpoonright l$ has been defined we let $\delta_s(l)$ be the least $k < s$ such that $\#V_k^l$ has increased since the last visit to $\delta_s \upharpoonright l$ (we let $\delta_s(l) = s$ if no $k < s$ is found).

(1.2) We initialize every node σ to the right of δ_s. For each node $\sigma \subset \delta_s$ we remove every σ clique and every σ link which has lower priority than δ_s.

(1.3) Next we update I_k^σ for each $\sigma \subset \delta_s$. This is again done inductively as follows. Suppose that the intervals for σ have been updated. We update the intervals for the node $\sigma * k \subset \delta_s$. If $I_k^\sigma = \emptyset$ then we must also have $I_m^{\sigma*k} = \emptyset$ for every m, in which case we do nothing here for $\sigma * k$. Otherwise assume that $I_k^\sigma \neq \emptyset$ (in which case $\max I_k^\sigma = s$). Now let m be such that $\sigma * k * m \subseteq \delta_s$. Set $I_n^{\sigma*k} = \emptyset$ for every $n > m$. If $I_m^{\sigma*k} \neq \emptyset$ we increase the right end-point of $I_m^{\sigma*k}$ to s. Otherwise if $I_m^{\sigma*k} = \emptyset$ we let $m' \leq m$ be the least such that $I_{m'}^{\sigma*k} = \emptyset$, and in this case set $I_{m'}^{\sigma*k} = (\max I_{m'-1}^{\sigma*k}, s]$. Finally if $m' = 0$ we set $I_0^{\sigma*k} = I_k^\sigma$.

Phase 2, Acting for each $\sigma \subset \delta_s$. For each node $\sigma \subset \delta_s$ where $\sigma * k \subseteq \delta_s$ we do the following (unless $I_0^\sigma = \emptyset$, in which case we do nothing for σ).

(2.1) *Forming new σ cliques.* For each $j \in \cup_n I_n^\sigma$ we say that j is currently eligible for (2.1) if j is not a member of any σ clique, $\ell_\sigma(j)$ exists, $g_\sigma(\ell_\sigma(j))[s] = \infty$ and $\#R_j$ has grown since the last time j was eligible for (2.1) with respect to σ and the class $M_{\ell_\sigma(j)}$. Find the least eligible $j \in \cup_n I_n^\sigma$ such that the current stage is the qth time j has been determined to be eligible for (2.1) with respect to σ and the class $M_{\ell_\sigma(j)}$, where q is even.

We form a new σ clique consisting of all $j \leq j' < \#M_{\ell_\sigma(j)}[s]$ such that j' is not currently a member of any σ clique and $\#R_{j'}[s] \geq \#R_j[s]$ and $j' \in \cup_n I_n^\sigma$ and $\ell_\sigma(j) = \ell_\sigma(j')$. Set $\ell(\mathcal{C}) = \ell_\sigma(j)$. Remove each σ link $\ell_\sigma(j')$.

Repeat with the next $j_1 > j$ and $j_1 \in \cup_n I_n^\sigma$ in place of j, forming a new σ clique with j_1 as the least element in the same way. Continue this way until all eligible elements (with even q) of $\cup_n I_n^\sigma$ have been exhausted.

(2.2) *Growing existing σ cliques.* For each σ clique $\mathcal{C}$ in existence and each $\min \mathcal{C} < j < \#M_{\ell(\mathcal{C})}[s]$ such that j is not currently a member of any σ clique and $\#R_j[s] \geq \operatorname{size} \mathcal{C}[s]$ and $j \in \cup_n I_n^\sigma$ and $\ell(\mathcal{C}) = \ell_\sigma(j)$ we add j to $\mathcal{C}$. Remove each $\ell_\sigma(j)$.

(2.3) *Dissolving σ cliques due to a g_σ change.* For each σ clique $\mathcal{C}$ in existence such that $g_\sigma(\ell(\mathcal{C}))[s] = f$ we do the following: Restore the σ link to $\min \mathcal{C}$ by setting $\ell_\sigma(\min \mathcal{C}) = \ell(\mathcal{C})$ and remove $\mathcal{C}$.

(2.4) *Updating obsolete σ links.* Go through each σ linked class R_i, starting with the smallest i, and for each such class we do the following.

(i) For every $i' > i$ such that $R_{i'}$ is σ linked and $\ell_\sigma(i') = \ell_\sigma(i)$ and $\#R_{i'}[s] \neq \#R_i[s]$ we remove the link $\ell_\sigma(i')$.

(ii) See if there exists i' and τ such that $R_{i'}$ is τ linked, $\ell_\tau(i') = \ell_\sigma(i)$, $\ell_\tau(i')$ has higher priority than $\ell_\sigma(i)$ and $\#R_{i'}[s] < \#R_i[s]$. If i' and τ exists we remove the link $\ell_\sigma(i)$.

(iii) See if there exists a τ clique $\mathcal{C}$ such that $\ell(\mathcal{C}) = \ell_\sigma(i)$, $\mathcal{C}$ has higher priority than $\ell_\sigma(i)$ and $\operatorname{size} \mathcal{C}[s] < \#R_i[s]$. If τ and $\mathcal{C}$ exists we remove the link $\ell_\sigma(i)$.

(iv) Finally if $\ell_\sigma(i)$ has not been removed by (ii) or (iii), we will remove every lower priority clique or link which disagrees with $\ell_\sigma(i)$. This is achieved by the following. For each i' and τ such that $R_{i'}$

is τ linked, $\ell_\tau(i') = \ell_\sigma(i)$, $\ell_\tau(i')$ has lower priority than $\ell_\sigma(i)$ and $\#R_{i'}[s] < \#R_i[s]$, we remove the link $\ell_\tau(i')$. For each τ clique $\mathcal{C}$ such that $\ell(\mathcal{C}) = \ell_\sigma(i)$, $\mathcal{C}$ has lower priority than $\ell_\sigma(i)$ and $\operatorname{size}\mathcal{C}[s] < \#R_i[s]$, we remove the clique $\mathcal{C}$.

Phase 3, Growing classes in M. For each finite class M_n of M we grow M_n (if necessary) to have the same size as $\min\{\#R_i[s] \mid \ell_\sigma(i) = n$ for some σ or $\ell(\mathcal{C}) = n$ for some clique $\mathcal{C}$ where $i \in \mathcal{C}\}$. If M_n has no link or clique pointing at it we declare $\#M_n = \infty$.

Phase 4, Resolving conflicts in M. For each x look at the collection of M classes $M_{n_0}, \cdots, M_{n_j}$ such that $\#M_{n_0} = \cdots = \#M_{n_j} = x$. Pick $m \leq j$ so that there is a clique or a link pointing at M_{n_m} which is of the highest priority (amongst all objects pointing at one of $M_{n_0}, \cdots, M_{n_j}$).

Declare $\#M_n = \infty$ for every $n \in \{n_0, \cdots, n_j\} - \{n_m\}$. We need to reassign the links which were pointing at one of these classes M_n that we have just declared to be infinite: If $\ell_\sigma(i) = n \in \{n_0, \cdots, n_j\} - \{n_m\}$ we remove $\ell_\sigma(i)$ and form a new link $\ell_\sigma(i) = n_m$. If $\ell(\mathcal{C}) = n \in \{n_0, \cdots, n_j\} - \{n_m\}$ where $\mathcal{C}$ is a σ clique, we form the link $\ell_\sigma(\min \mathcal{C}) = n_m$ and remove the clique $\mathcal{C}$.

Phase 5, Establishing new σ links for each $\sigma \subset \delta_s$. For each $\sigma \subset \delta_s$ and each $i \in \cup_n I_n^\sigma$ where i is not a member of any σ clique, i is not σ linked and $\#R_i > i$ we will place a link $\ell_\sigma(i)$. If there is already an M-class M_n such that $\#M_n = \#R_i$ we declare $\ell_\sigma(i) = n$, otherwise we introduce a new M-class (by picking the least n such that M_n has not been used and setting $\#M_n = \#R_i$) and declare $\ell_\sigma(i) = n$.

Note that $\emptyset'$ can tell, for any given class R_i, whether $\#R_i \leq i$; if so then no link will be formed for R_i but in this case we know that $\#R_i$ is finite.

3.5. Verification

Lemma 3.6. *At each point in the construction, for each σ and i, there can be either a unique σ link on R_i, or a unique σ clique containing R_i, possibly neither, but never both.*

Proof. Straightforward examination of the construction. □

Lemma 3.7. *If a link or a clique ℓ_0 is of higher priority than another ℓ_1, then this stays true until one of the two is cancelled.*

Proof. Suppose that ℓ_0 is associated with σ and ℓ_1 with τ. The only non-trivial case to check is when $\sigma \supseteq \tau * k$ for some k (or vice versa). In either case ℓ_0 remains of higher priority than ℓ_1 unless $\min I_k^\tau$ changes. Under (1.3) of the construction, $\min I_k^\tau$ changes only if the construction visits left of $\tau * k$, which means that σ would be initialized. □

Lemma 3.8. *Suppose at some stage s, R_i is σ linked where $\ell = \ell_\sigma(i)$, or R_i is part of a σ clique $\mathcal{C}$ where $\ell = \ell(\mathcal{C})$. Then $\#M_\ell[s] > i$.*

Proof. Fix i and σ. We argue by induction on s. By Lemma 3.6 at each point of the construction we only need to consider either a σ link on R_i or a σ clique $\mathcal{C}$ on R_i.

A σ link $\ell = \ell_\sigma(i)$ can be formed under Phase (2.3), 4 or 5. When a σ link $\ell = \ell_\sigma(i)$ is first formed under Phase 5 we certainly have $\#M_\ell = \#R_i > i$. If it is formed under Phase 4 then the new target M-class has the same size as the old. Lastly if ℓ is formed under (2.3) we apply the induction hypothesis.

Class R_i will join a σ clique $\mathcal{C}$ under (2.1) or (2.2); we can simply check each case (we apply the induction hypothesis for (2.1)). Note that $\ell(\mathcal{C})$ is never retargetted until $\mathcal{C}$ is removed. □

Lemma 3.9. *Suppose that M_n is a non-empty class. Then at the end of every stage, M_n is finite iff there is at least one link or clique pointing at M_n.*

Proof. If M_n is declared infinite under Phase 3 or 4 then all links and cliques pointing at M_n are removed immediately. No link or clique can afterwards be made to point at the infinite class M_n (the only action which creates a new link is in Phase 5, which only targets finite classes).

Now conversely if there are no links or cliques pointing at M_n we would declare $\#M_n = \infty$ in the same stage under Phase 3. □

Lemma 3.10. *At every point of the construction where $M_n \neq \emptyset$ we have $\#M_n \leq \#R_i$ for every i which is involved in a link or a clique pointing at M_n.*

Proof. By Lemma 3.9 we can assume that M_n is finite. When M_n is first used under Phase 5 it was set equal in size to the only R class pointing at it. Thereafter if a new link or clique is formed pointing at M_n (2.1) or (2.3) we apply the induction hypothesis. If a clique picks up a new element R_j under (2.2) then we also apply induction hypothesis. Under Phase 3 we never grow M_n beyond the minimum size. Under Phase 4 we apply the induction hypothesis. Phase 5 is obvious by construction. □

Lemma 3.11. *Suppose at stage s of the construction a node τ is visited. If a τ link $\ell_\tau(i)$ exists after Phase 3 is done, then $\#M_{\ell_\tau(i)} = \#R_i$.*

Proof. In Phase 2 of the construction at stage s we act for τ. In particular under (2.4) we ensure that any object $\ell_{\tau'}(i')$ or $\ell(\mathcal{C})$ also pointing at $M_{\ell_\tau(i)}$ has got $\#R_{i'} \geq \#R_i$ or size$\,\mathcal{C} \geq \#R_i$. □

Lemma 3.12. *M is an equivalence structure on a subset of the condensation $\widehat{R}$.*

Proof. We may assume that M has infinitely many infinite classes. There are two things to check. First, we need to verify that no two finite classes of M are equal in size. This is explicitly ensured by Phase 4 of the construction.

Second, we need to check that if $\#M_n < \infty$ then there is some R-class with the same size. Assume that $\#M_n = x$. Consider a stage s large enough so that the classes M_n and $R_0, \cdots, R_{x-1}$ are all stable. This means that for $i < x$, if R_i is finite then it does not increase in size after stage s and if R_i is infinite then $\#R_i[s] > x$. By the construction Phase 3, Lemmas 3.9 and 3.10 we would have that $x = \#M_n = \#R_i$ for some class R_i currently pointing at M_n. By Lemma 3.8, $i < x$, and since $\#R_i[s]$ is stable at stage s, we conclude that R_i has size x. □

Lemma 3.12 together with Lemma 3.1 tells us that there is some l such that $\lim_s g_l(x,s)$ is a guessing function for structure M. (Note that Lemma 3.1 applies even if M has only finitely many finite sizes; hence there is no need to explicitly ensure during the construction that M has infinitely many finite sizes.)

We define the true node of construction, σ_{true}, to be the leftmost node with the property that:

- σ_{true} is visited infinitely often, and
- for every k, $\sigma_{true} * k$ is visited finitely often.

This node σ_{true} exists because some $\lim_s g_l(x,s)$ is a guessing function for the structure M.[3]

Lemma 3.13. *There are only finitely many stages such that the construction visits left of σ_{true}.*

Proof. Note that every node τ where $|\tau| = |\sigma_{true}|$ must have Π^0_3-outcome (if τ is visited infinitely often). Thus the set of all nodes which are visited infinitely often is a well-founded tree. Hence if there are infinitely many stages s where δ_s moves left of σ_{true} then this would contradict the choice of σ_{true}. □

Lemma 3.14. *Suppose that $\#R_i = \infty$. Then there is an s such that for every $j \in \Xi(i,s)$, the class R_j is infinite.*

Proof. Suppose the contrary that for every s there is a number $j_s \in \Xi(i,s)$ such that $\#R_{j_s} < \infty$. Pick $j_s \in \Xi(i,s)$ such that $\#R_{j_s}$ is least. Define the function $f(s,t) = 0$ if $t < s$ and equal to $\min\{\#R_k[t] \mid k \in \Xi(i,s)[t]\}$ if $t \geq s$. It is easy to check that $f(s,t)$ is computable and witnesses that the function $\hat{f}(s) = \lim_t f(s,t)$ is limitwise monotonic. (The fact that $\hat{f}(s)$ is defined follows from the fact that $f(s,t) \leq \#R_{j_s}[t]$ at every stage t after which j_s is enumerated in $\Xi(i,s)$.)

Fix s and let t_0 be a stage after which $f(s,t)$ is stable, say with value c.

Claim 3.15. *There is some $k \in \Xi(i,s)$ such that $\#R_k = c$.*

Proof of claim. Let $t_1 > t_0$ be a stage where for every $k \in \Xi(i,s)[t_0]$, we have $\#R_k[t_1] > c$. (If t_1 does not exist then we are done.) Now let $t_2 > t_1$ be such that for every $k \in \Xi(i,s)[t_1]$, we have $\#R_k[t_2] > c$. We also require that for every $j < c$, we have $\#R_j[t_2] > c$ or R_j finally has size $< c$. Clearly if t_2 does not exist we are done, so we assume for a contradiction that t_2 exists.

We argue that if a class R_j is added to $\Xi(i,s)$ at some stage u between t_1 and t_2 then either $j < c$ or $\#R_j[u] > c$. Suppose this is false as witnessed by R_j and a least u. At stage u we must have $\#R_j[u] = \#R_k[u] = c$ for some k already in $\Xi(i,s)$. Since $j \geq \#R_k[u]$ we see that $\#R_k$ must have grown since k was enumerated in $\Xi(i,s)$. This means that k cannot have been enumerated into $\Xi(i,s)$ after t_0 (because $f(i,-) = c$ after t_0). But if k was enumerated into $\Xi(i,s)$ before t_0 then we have $\#R_k[u] > c$ by the choice of t_1. In either case we get a contradiction. Thus we conclude that if a class R_j is added to $\Xi(i,s)$ at some stage u between t_1 and t_2 then either $j < c$ or $\#R_j[u] > c$.

Now at stage t_2 we have $f(s,t_2) = \min\{\#R_k[t_2] \mid k \in \Xi(i,s)[t_2]\} = c$. Let j be such that $\#R_j[t_2] = c$. Then clearly j must have been enumerated in $\Xi(i,s)$ at some stage u where $t_1 < u \leq t_2$. By the preceding paragraph we have that either $j < c$ or $\#R_j[u] > c$. Both alternatives are clearly impossible (the first alternative contradicts the choice of t_2). This contradiction shows that t_2 cannot exist and so the claim is proved. □

Claim 3.15 says that for every s, $\hat{f}(s)$ is a size of a finite class of the condensation $\widehat{R}$. Furthermore by the definition of $\Xi(i,s)$, for every $k \neq i$, $k \in \Xi(i,s)$, we must have that the final size of $\#R_k$ is no smaller than $\#R_i[s]$. By the assumption that $\#R_i = \infty$, the range of $\hat{f}$ is an infinite subset of the finite sizes of the condensation $\widehat{R}$. Since we assumed that $\widehat{R}$ is categorical we obtain a contradiction by applying Theorem 4.1(i) and Lemma 3.1. □

Fix a stage s_{true} large enough so that after stage s the construction never visits left of σ_{true}. We also assume that for every $j < \min I_0^{\sigma_{true}}$ such that $\#R_j < \infty$, the class R_j is stable after s_{true}. We also assume

[3] The Recursion Theorem is used here to provide an index for M; its use can be avoided by directly monitoring each index l and comparing it against the partially built structure M (this still involves a similar Π^0_3 guessing procedure and a similar setup). However this latter approach is more cumbersome.

that s_{true} is large enough so that for every $j < \min I_0^{\sigma_{true}}$ such that $\#R_j = \infty$, the stage s_{true} is larger than the s given in Lemma 3.14, and that $R_j \neq \emptyset$.

Definition 3.16. Let

$$\begin{aligned} \widehat{J} &= \left\{ j : j < \min I_0^{\sigma_{true}} \text{ such that } \#R_j = \infty \right\}, \\ \widehat{\Xi} &= \bigcup \left\{ \Xi(j, s_{true}) : j \in \widehat{J} \right\}, \\ X_0 &= \max \left\{ \#R_j \mid j < \min I_0^{\sigma_{true}} \text{ such that } \#R_j < \infty \right\}. \end{aligned}$$

It is easy to check that $\Xi(j, s) \supseteq \Xi(j, s+1)$ for every j, s, and hence for every $k \in \widehat{\Xi}$, the class R_k is infinite. We call ℓ a *(τ, j) object* if $\ell = \ell_\tau(j)$ or if $\ell = \ell(\mathcal{C})$ where $\mathcal{C}$ is a τ clique such that $\min \mathcal{C} = j$. If a link is replaced by a new one in a single step (for instance, under (2.1), (2.3) or Phase 4) we consider it to be a different object.

Given a (τ, j) object ℓ existing at stage s, we define the *origin of ℓ* to be the greatest stage $s' < s$ such that a (τ, j) object ℓ' is formed under Phase 5 at s'.

Lemma 3.17. *Fix τ, j. If ℓ_0 is a (τ, j) object existing at stage s_0 and ℓ_1 is a (τ, j) object existing at stage $s_1 > s_0$ with the same origin, then $\#M_{\ell_0}[s_0] \leq \#M_{\ell_1}[s_1]$. In particular, if ℓ_0 has origin s then $\#M_{\ell_0}[s_0] \geq \#R_j[s]$.*

Proof. Each (τ, j) object ℓ can be traced back to the stage when it was formed. This has to be under steps (2.1), (2.3), Phase 4 or Phase 5 of the construction. In the first three cases we are not yet at the origin, and can continue with another ℓ' with the same origin as ℓ. In the first two cases (2.1), (2.2) we in fact have $\ell = \ell'$ while in the third case (Phase 4) we also have $\#M_\ell = \#M_{\ell'}$.

The last statement in the lemma follows easily from the previous, by following ℓ_0 to its origin. □

Lemma 3.18. *Let ℓ be a (τ, j) object existing at stage $t_0 > s_{true}$, where τ, j is any pair. Suppose that at stage t_0, every i_0 pointing at (or linked to) M_ℓ is in $\widehat{\Xi}$. Then for every i, if i is involved in an object made to point at M_ℓ after stage t_0 (before ℓ is killed), we will also have $i \in \widehat{\Xi}$.*

Proof. i is made to point at M_ℓ when a new (τ', i) object ℓ' involving i is formed such that $M_\ell = M_{\ell'}$, or when i joins a clique. Let's consider each case separately.

The new link ℓ' can only be formed under Phase 4 or Phase 5 of the construction. Note that steps (2.1) and (2.3) are not possible since no object involving i was pointing at M_ℓ at t_0. So suppose ℓ' was formed under Phase 4 or 5 at stage $t > s_{true}$. By Phase 3 of the construction, there is some class i_0 pointing at M_ℓ such that $\#R_{i_0} = \#M_\ell$. Thus $\#R_i[t] \geq \#M_\ell[t] = \#R_{i_0}[t]$, and by Lemma 3.8 we have that $\#R_{i_0}[t] > i$. Since i_0 is already a member of $\widehat{\Xi}$ at stage t, we conclude by Definition 3.5 that $i \in \widehat{\Xi}$.

Next we assume that i joins a clique $\mathcal{C}$ pointing at M_ℓ at stage $t > s_{true}$ under (2.1) or (2.2). This means that $\#R_i[t] \geq \#R_{i_0}[t]$ for some class i_0 already pointing at M_ℓ. Now by the construction and by Lemma 3.10 we see that $i < \#M_\ell[t] \leq \#R_{i_0}[t]$. Since i_0 is already a member of $\widehat{\Xi}$ at stage t, we conclude that $i \in \widehat{\Xi}$. □

Lemma 3.19. *Let ℓ_0 be a (τ, j) object existing at stage $t_0 > s_{true}$, and ℓ_1 be a (τ, j) object formed after ℓ_0 with the same origin as ℓ_0, where τ, j is any pair. Suppose that at stage t_0, every i_0 pointing at M_{ℓ_0} is in $\widehat{\Xi}$. Then between the time when ℓ_1 is formed until the time when ℓ_1 is killed, if i is pointing at M_{ℓ_1} then $i \in \widehat{\Xi}$.*

Proof. We fix ℓ_0 and apply induction on the formation of ℓ_1. ℓ_1 is formed under (2.1), (2.3) or Phase 4. In the first two cases (2.1), (2.3) it is easy to see that Lemma 3.19 follows by applying a combination of the induction hypothesis, Lemma 3.18 and Definition 3.5.

Let's assume that ℓ_1 is formed after ℓ_0 via Phase 4. Now ℓ_1 must have been redirected from another (τ, j) object ℓ_1', where ℓ_1' was replaced by ℓ_1 (note that ℓ_1' could have been a τ clique). Let k_0 be pointing at $M_{\ell_1'}$ and such that $\#R_{k_0} = \#M_{\ell_1'}$. By induction hypothesis for ℓ_1' we have $k_0 \in \widehat{\Xi}$. However in order for Phase 4 to apply and produce ℓ_1 we must have $\#M_{\ell_1} = \#M_{\ell_1'}$ which means that for every i pointing at M_{ℓ_1} we have $\#R_i \geq \#R_{k_0}$. (Note that this includes all the classes pointing at some M_n, where $\#M_n = \#M_{\ell_1}$ and which is redirected to point at M_{ℓ_1}.) By Lemma 3.8, $i < \#R_{k_0}$ and so we have i is in $\widehat{\Xi}$. So every class i pointing at M_{ℓ_1} is in $\widehat{\Xi}$. □

Lemma 3.20. *Let $j \in \widehat{\Xi}$ and $\tau \subseteq \sigma_{true}$. For almost every (τ, j) object ℓ we have that $i \in \widehat{\Xi}$ if i is also (involved in a link or clique) pointing at M_ℓ at the same time as ℓ.*

Proof. Fix τ and j as in the statement of the lemma. For ease of notation we assume that at stage s_{true}, j has entered $\widehat{\Xi}$ and R_j has grown in size since it entered $\widehat{\Xi}$. We let ℓ range over all (τ, j) objects. We assume that there are infinitely many (τ, j) objects (otherwise it is trivial). Suppose ℓ_0 is a (τ, j) object formed at $t > s_{true}$ under Phase 5. Then $\#R_j[t] = \#M_{\ell_0}[t]$ and by Definition 3.5 and Lemma 3.10 any i also pointing at M_{ℓ_0} at t will be enumerated in $\widehat{\Xi}$ at t. Now if some ℓ_1 is formed after s_{true} under Phase 5, then almost every ℓ formed after s_{true} has the same origin as an ℓ_1 formed under Phase 5, and in this case we apply Lemma 3.19 to obtain Lemma 3.20.

Suppose that at the end of Phase 3 at some stage $t > s_{true}$ in which τ is visited, we have a τ link $\ell_0 = \ell_\tau(j)$ in existence. By Lemma 3.11 we have $\#M_{\ell_0} = \#R_j$. This means that at the end of Phase 3, any class i pointing at M_{ℓ_0} must be in $\widehat{\Xi}$. Applying Lemma 3.19, we also obtain Lemma 3.20.

So finally we only need to consider the situation where:

- No ℓ is formed under Phase 5 after s_{true}.
- At the end of Phase 3 during a stage $t > s_{true}$ in which τ is visited, we have a (τ, j) object in existence which must be a τ clique.

We show that this situation is impossible. These two assumptions imply that no (τ, j) object ℓ_0 can be formed under (2.3) at $t > s_{true}$: Otherwise this ℓ_0 must remain after its formation until the end of Phase 3 in the same stage t, which contradicts the second assumption above. This means that the (τ, j) objects formed after s_{true} are formed alternately by (2.1) and Phase 4.

Notice that (2.1) must be applied infinitely often, because otherwise the only (τ, j) objects which can eventually exist are τ links, and since τ is visited infinitely often we get a contradiction to the second assumption. We claim that there is no stage $t > s_{true}$ in which τ is visited and j is determined to be eligible for (2.1) with respect to τ and $M_{\ell_\tau(j)}[t]$ for the qth time for some odd q: If there is such a stage t then the link $\ell_\tau(j)$ must remain until the end of Phase 3 of stage t, contradicting the second assumption above. Since no such stage t exists, each time j is determined to be eligible for (2.1) after stage s_{true}, it has to be with respect to some $M_{\ell_\tau(j)}$ which was first determined eligible before stage s_{true}. Since there are only finitely many such M-classes, one of them has to be applied to (2.1) infinitely often, which means that it has to be determined eligible an odd number of times after s_{true}, contradicting an earlier statement in this paragraph. This final contradiction ends the proof of Lemma 3.20. □

Lemma 3.21. *Given any x, j and $\tau \subseteq \sigma_{true}$ such that $\#R_j > x$, there are only finitely many (τ, j) objects ℓ such that when ℓ is formed, $\#M_\ell \leq x$.*

Proof. Suppose there are infinitely many (τ, j) objects ℓ with $\#M_\ell \leq x$. Only steps (2.1), (2.3), Phase 4 and 5 of the construction will cause a new (τ, j) object ℓ to be formed, so infinitely many ℓ with $\#M_\ell \leq x$ will have to be formed at one of these steps. We say that (2.1), (2.3), Phase 4 or Phase 5 applies at stage s if this is the situation under which an ℓ is formed at a stage s.

First observe that only finitely many ℓ can be formed at Phase 5 (i.e. Phase 5 applies finitely often), because otherwise for every ℓ formed late enough under Phase 5, we have $\#M_\ell = \#R_j > x$, and we can apply Lemma 3.17. So we may now assume that every (τ, j) object formed after s_{true} have the same origin.

We proceed similarly as in the proof of Lemma 3.20. Suppose that at the end of Phase 3 at some stage $t > s_{true}$ in which τ is visited, we have a τ link $\ell_0 = \ell_\tau(j)$ in existence. By Lemma 3.11 we have $\#M_{\ell_0} = \#R_j > x$, and we can again apply Lemma 3.17. Thus we will only need to consider the situation where:

- No ℓ is formed under Phase 5 after s_{true}.
- At the end of Phase 3 at each stage $t > s_{true}$ in which τ is visited, we have a (τ, j) object in existence which is a τ clique.

The rest of the proof follows exactly the proof of Lemma 3.20. □

Lemma 3.22. *For every* $i \geq \min I_0^{\sigma_{true}}$ *such that* $\max\{i, X_0\} < \#R_i$, *one of the following holds:*

- $i \in \widehat{\Xi}$.
- *There is a stable* σ_{true} *link* $\ell_{\sigma_{true}}(i)$.
- *There is a stable* σ_{true} *clique* $\mathcal{C}$ *containing* i.

Proof. We proceed by induction on $i \geq \min I_0^{\sigma_{true}}$. Assume the statement of the lemma holds for every $k < i$. We consider a stage $s^* > s_{true}$ large enough so that for each k such that $\min I_0^{\sigma_{true}} \leq k < i$ and $\#R_k > k, X_0$, if $k \in \widehat{\Xi}$ then R_k has grown in size since k was enumerated in $\widehat{\Xi}$, and if $k \notin \widehat{\Xi}$ it is already involved in a stable link or clique. Suppose that there is no stable σ_{true} link and no stable σ_{true} clique $\mathcal{C}$ containing i. We argue that $i \in \widehat{\Xi}$.

Since $\#R_i > i$ there are infinitely many stages where i is involved in a σ_{true} object. Suppose that there are infinitely many stages where i is made to join a σ_{true} clique under (2.1) or (2.2) where $k = \min \mathcal{C} < i$. By Lemmas 3.8 and 3.10, $\#R_k > k$. Since these cliques have to be removed after i joins (else i is permanently part of a σ_{true} clique), we have that infinitely many (σ_{true}, k) cliques are formed under (2.1). In that case it is straightforward to verify that $\#R_k = \infty > X_0$ and we may apply the induction hypothesis for k. By the choice of s^*, we have that $k \in \widehat{\Xi}$. Since there are necessarily infinitely many (σ_{true}, k) cliques, by Lemma 3.20 we will also have $i \in \widehat{\Xi}$.

So we may now assume it is the case that almost every σ_{true} object which i is involved in is a (σ_{true}, i) object. Since every outcome of σ_{true} is visited finitely often, (1.2) can only apply finitely often to remove a (σ_{true}, i) object. So each (σ_{true}, i) object will have to be removed under (2.1), (2.3), (2.4) or Phase 4.

Case (2.4). We first consider (2.4). Suppose (2.4) applies infinitely often to remove a (σ_{true}, i) object. Each time (2.4) applies there is a (τ, k) object ℓ' and a (σ_{true}, i) object ℓ such that $M_{\ell'} = M_\ell$ and ℓ' is of higher priority. Let's begin by fixing a (τ, k) object ℓ' which is never removed, and suppose there are infinitely many stages such that $M_\ell = M_{\ell'}$, where ℓ is a (σ_{true}, i) object of lower priority, and where (2.4) applies to remove ℓ.

Suppose that $\#M_{\ell'} = x < \infty$. By Lemmas 3.10 and 3.21, $\#R_i = x$. This means that (2.4)(ii) and (2.4)(iii) cannot apply. If (2.4)(i) were to apply then we would have $k < i$ and $\tau = \sigma_{true}$ and $\ell' = \ell_{\sigma_{true}}(k)$, such that $\#R_k > x$. By Lemma 3.11 this link $\ell_{\sigma_{true}}(k)$ must be killed at every such stage where (2.4)(i) applies. So (2.4)(i) cannot apply infinitely often for ℓ'.

Finally we consider (2.4)(iv). If this were to apply then $\ell' = \ell_\tau(k)$ is a link and $\#R_k > x$. This follows from the fact that $\#R_i = x$ and if $\mathcal{C}$ is a (σ_{true}, i) clique pointing at $M_{\ell'}$ then size $\mathcal{C} = x$. Again by Lemma 3.11 this link $\ell' = \ell_\tau(k)$ must be killed at every such stage where (2.4)(iv) applies (since τ is visited at each such stage). Hence (2.4)(iv) cannot apply infinitely often for ℓ' as well. So the case $M_{\ell'} < \infty$ is impossible.

Claim 3.23. *Fix a class M_y such that $\# M_y = \infty$, and assume that there are infinitely many stages such that $M_y = M_\ell$ for some (σ_{true}, i) object ℓ. If infinitely many of these objects ℓ are removed under (2.4)(iv) then $i \in \widehat{\Xi}$.*

Proof of claim. Suppose that (2.4)(iv) is applied infinitely often in which some (τ', k') link $\ell_{\tau'}(k')$ kills some ℓ where $M_\ell = M_y$. Since τ' has to be visited infinitely often and be of higher priority, there are only finitely many possibilities for τ', namely, $\tau' \subseteq \sigma_{true}$, so we fix a τ' which is infinitely often responsible.

If $\tau' = \sigma_{true}$ then we must have $k' < i$ such that $M_{\ell_{\tau'}(k')} = M_\ell = M_y$ infinitely often. Hence $\# R_{k'} = \infty > k', X_0$ and so we can apply the induction hypothesis for k', to obtain that either $k' \in \widehat{\Xi}$ or that k' is already involved in a stable σ_{true} link or clique. Since $g_{\tau'}(y) = g_{\sigma_{true}}(y)$ is eventually a stable ∞, therefore, if k' is already involved in a stable σ_{true} link or clique at s^* then this stable object must be a σ_{true} clique, which means that the link $\ell_{\tau'}(k')$ cannot exist to kill ℓ infinitely often. Hence, we see that k' cannot be involved in a stable object at s^*, which means, by the choice of s^*, that infinitely many (σ_{true}, k') objects must exist during the construction and that $k' \in \widehat{\Xi}$. By Lemma 3.20 we will also have $i \in \widehat{\Xi}$.

On the other hand suppose that $\tau' \subset \sigma_{true}$, then $k' < \min I_0^{\sigma_{true}}$. Since $\# R_{k'} \geq \# M_{\ell_{\tau'}(k')} = \infty$, hence $k' \in \widehat{J}$ and so $k' \in \widehat{\Xi}$. We would like to apply Lemma 3.20 to conclude that $i \in \widehat{\Xi}$, unfortunately this cannot be done unless we know that there are infinitely many different (τ', k') objects. Suppose this is not the case; so there is a final stable link $\ell_{\tau'}(k')$ which is never removed. By Lemma 3.11, and since τ' is visited infinitely often, we see that at some large stage $t_0 > s^*$, we have that every i_0 pointing at $M_{\ell_{\tau'}(k)}$ is also in $\widehat{\Xi}$. By Lemma 3.18 we conclude that $i \in \widehat{\Xi}$. □

Now we suppose that $\# M_{\ell'} = \infty$. Since σ_{true} is the true node, we must eventually have the stable value $g_{\sigma_{true}}(\ell') = \infty$. By assumption there are infinitely many stages such that the (τ, k) object ℓ' kills some (σ_{true}, i) object under (2.4). If infinitely many of these steps are under (2.4)(iv) then we apply Claim 3.23 to get $i \in \widehat{\Xi}$. Suppose only finitely many of these are under (2.4)(iv). This means that ℓ' will infinitely often remove some ℓ under (2.4)(i), (ii) or (iii). We claim that there are infinitely many stages in which a (σ_{true}, i) clique $\ell(\mathcal{C})$ is pointing at $M_{\ell'}$: Suppose not. Since ℓ' will infinitely often remove some ℓ under (2.4)(i), (ii) or (iii), this means that there are infinitely many stages in which σ_{true} is visited and some link $\ell_{\sigma_{true}}(i)$ exists and is pointing at $M_{\ell'}$ (note that this link $\ell_{\sigma_{true}}(i)$ cannot be formed at step (2.3) of the same stage, because $g_{\sigma_{true}}(\ell') = \infty$). By the assumption that almost every σ_{true} object that i is involved in is a (σ_{true}, i) object, we see that i never joins a σ_{true} clique with least element $< i$. Hence eventually we must apply (2.1) to get a (σ_{true}, i) clique $\mathcal{C}$ such that $\ell(\mathcal{C})$ is pointing at $M_{\ell'}$, a contradiction.

Thus there must be infinitely many stages in which a (σ_{true}, i) clique $\ell(\mathcal{C})$ is pointing at $M_{\ell'}$. Each clique has to be removed after it is formed, since i is never part of a stable link or clique; how can each such clique be removed? It is easy to see that out of the possibilities (2.1), (2.3), (2.4) or Phase 4, only (2.4)(iv) is possible. (Phase 4 is not possible because otherwise ℓ' is destroyed along with $\ell(\mathcal{C})$.) In this case we apply Claim 3.23 to get $i \in \widehat{\Xi}$.

We now conclude that if there exists some (τ, k) object ℓ' which is never removed, and which infinitely often removes some (σ_{true}, i) object under (2.4), we have that $i \in \widehat{\Xi}$. Since there are only finitely many ℓ' formed before stage s^*, we may henceforth assume that each ℓ' responsible for killing some ℓ under (2.4) is formed after s^*.

Let Y be such that $i \in I_Y^{\sigma_{true}}$. Since there are only finitely many pairs (τ, k), τ to the left of $\sigma_{true} * (Y+1)$, where a (τ, k) object is formed during the construction, we now assume that s^* is large enough so that after s^*:

- If some (τ, k) object is formed after s^*, where τ is to the left of $\sigma_{true} * (Y+1)$, then there are infinitely many different (τ, k) objects formed during the construction.

- No (τ, k) object τ to the left of $\sigma_{true} * (Y + 1)$ is removed under (2.4) after s^*. (This is because τ is never again visited and so if enough (τ, k) objects are removed under (2.4) there will be no further (τ, k) objects.)

Claim 3.24. *No (τ, k) object is formed after s^*, where τ is to the left of $\sigma_{true} * (Y + 1)$ and k is any number.*

Proof of claim. We say that (τ, k) is a stable pair if either τ is to the left of $\sigma_{true} * (Y + 1)$ or $\tau \subseteq \sigma_{true}$ and $k < \min I^{\tau}_{Y_0}$ where $\tau * Y_0 \subseteq \sigma_{true} * Y$. By examining the proof of Lemma 3.13, we see that since we never visit left of a stable pair, the priority ordering between (τ, k) objects (for stable pairs (τ, k)) is completely determined by the pair (τ, k) at stage s^*. Furthermore it is straightforward (though tedious) to check that if there is a (τ'', k'') object of higher priority than a (τ, k) object for a stable pair (τ, k), then (τ'', k'') is also a stable pair.

For a contradiction let's fix a pair (τ, k) where τ is left of $\sigma_{true} * (Y + 1)$ with an object formed after s^*. By the choice of s^*, there are infinitely many (τ, k) objects formed after s^*. Each (τ, k) object must be formed under Phase 4 (all other actions require τ to be visited by the construction). Let $s^* < t_0 < t_1 < \cdots$ be the stages where this happens, and let M_{n_i} be the class which the new (τ, k) object formed at stage t_i is made to point at. Note that at every stage strictly in between t_i and t_{i+1}, there is a (τ, k) object pointing at M_{n_i}, and at stage t_{i+1} this (τ, k) object ℓ will get replaced by a new one pointing at $M_{n_{i+1}}$. Each time this happens there is a (τ', k') object ℓ' which was already pointing at $M_{n_{i+1}}$ before the action at t_{i+1}. Let us refer to this scenario as *ℓ' injuring ℓ at t_{i+1}*.

Since (τ, k) is a stable pair, by the first paragraph above, (τ', k') must also be a stable pair. It is straightforward to check that there are only finitely many stable pairs which has an associated object during the construction. So, let's fix a stable pair (τ', k') infinitely often responsible for injuring some ℓ. Since the priority of stable objects are determined by the pair, we fix a (τ', k') of the highest priority amongst the stable pairs infinitely often injuring some ℓ.

We begin by supposing that τ' is to the left of $\sigma_{true} * (Y + 1)$, and consider that the (τ', k') object ℓ' injures some ℓ at t_i. Since τ' is never again visited, observe that ℓ' cannot be removed strictly in between t_i and t_{i+1}: It cannot be removed by (2.4) by the assumptions on s^*, and it cannot be removed by Phase 4 because otherwise ℓ will be removed before t_{i+1}. Thus at stage t_{i+1} a new $\ell_{\tau'}(k')$ will be formed pointing at $M_{n_{i+1}}$. Continuing this way, we see that at every stage after t_i there is a (τ, k) object ℓ and a (τ', k') object ℓ' such that $M_\ell = M_{\ell'} = M_{n_j}$ for some j. This is a contradiction because some (τ', k') object ℓ' must after stage t_i injure some ℓ, and the two cannot point at the same M-class before the injury.

Thus we must have $\tau' \subseteq \sigma_{true}$, and we again consider that a (τ', k') object ℓ' injures some ℓ at t_i. By the argument in the preceding paragraph, there must exist infinitely many $j > i$ such that a (τ', k') object ℓ' is removed strictly between t_j and t_{j+1}.

This case is a bit trickier because τ' can now be visited by the construction infinitely often. Let's examine the possibilities for ℓ' to be removed strictly between t_j and t_{j+1}. Phase 4 is again not possible because otherwise ℓ will be removed before t_{j+1}. (2.3) is possible but we immediately replace ℓ' with another (τ', k') object pointing at the same M-class. If (2.1), (2.2) or (2.4) removes ℓ' then ℓ' is replaced by another higher priority object pointing at the same M-class. So we see that at each such j there is a (τ'', k'') object ℓ'' (of equal or of higher priority than ℓ') such that $M_{\ell''} = M_{\ell'} = M_\ell = M_{n_j}$.

Since ℓ'' has higher priority than ℓ, we see that (τ'', k'') must also be a stable pair. Now if ℓ'' itself was to be removed before stage t_{j+1} then we would have yet another stable pair (τ''', k''') with an object pointing at M_{n_j} of the same or of higher priority than ℓ''. In any case we must have the situation that just before the action Phase 4 at t_{j+1}, there is an object $\widehat{\ell}$ pointing at M_{n_j} where $\widehat{\ell}$ is of equal or of higher priority than (τ', k'). This means that in order for M_{n_j} to be killed by $M_{n_{j+1}}$ there must already be an object pointing at $M_{n_{j+1}}$ of higher priority than $\widehat{\ell}$ (and higher than (τ', k') as well) injuring some ℓ. Since this happens for infinitely many j, we get a contradiction to the assumption that (τ', k') has the highest priority amongst all stable pairs which infinitely often injures some ℓ. □

Claim 3.25. *For each τ, k and i, suppose there are infinitely many stages such that $M_\ell = M_{\ell'}$, where ℓ is a (σ_{true}, i) object and ℓ' is a (τ, k) object of higher priority formed after s^*, and $\tau \subseteq \sigma_{true}$. Then $i \in \widehat{\Xi}$.*

Proof of claim. We fix a $\tau \subseteq \sigma_{true}$ and a k such that there are infinitely many stages where a (τ, k) object ℓ' points at the same class as an ℓ.

If $\tau = \sigma_{true}$ then we must have $k < i$ such that $M_{\ell'} = M_\ell$ infinitely often. If $\#R_k = y < \infty$ then we claim that $\#R_i \leq y$: If not then $\#R_i > y$, and by Lemma 3.21 we see that only finitely many ℓ can have $\#M_\ell \leq y$. However since there are infinitely many different (σ_{true}, i) objects ℓ, we get a contradiction by applying Lemma 3.10 to get $\#M_\ell = \#M_{\ell'} \leq y$. Hence we see that $\#R_k > X_0, k$ (since $\#R_i > X_0, i$ by assumption). Thus we may apply the induction hypothesis for k. Since ℓ' is formed after s^*, by the choice of s^* there are infinitely many (τ, k) objects and $k \in \widehat{\Xi}$. Hence by Lemma 3.20 we will have $i \in \widehat{\Xi}$.

On the other hand if $\tau \subset \sigma_{true}$ then $k < \min I_0^{\sigma_{true}}$. If $\#R_k = \infty$ then $k \in \widehat{J}$ and $k \in \widehat{\Xi}$. Since s_{true} is large enough, we may assume that since some (τ, k) object is formed after s_{true}, there will be infinitely many different (τ, k) objects during the construction. Then by Lemma 3.20 we have $i \in \widehat{\Xi}$. So we must instead have $\#R_k = y < \infty$. Again by Lemmas 3.10 and 3.21 we see that $\#R_i \leq y = \#R_k$. But this is impossible since $\#R_i > X_0$. □

By Claim 3.25 we may assume that almost every ℓ' responsible for killing some ℓ under (2.4) is a τ object for some τ to the left of $\sigma_{true} * (Y + 1)$. Even though there are infinitely many nodes to the left of $\sigma_{true} * (Y + 1)$, only finitely many of them are ever visited by the construction, so we fix a τ to the left of $\sigma_{true} * (Y + 1)$ infinitely often responsible for killing ℓ under (2.4), and fix an associated k. Since any object responsible for killing ℓ under (2.4) is assumed to be formed after s^*, we get a contradiction by applying Claim 3.24.

This ends the analysis for case (2.4). Let's assume that (2.4) applies finitely often and we now consider the remaining cases (2.1), (2.3), Phase 4.

Case 4. Suppose Phase 4 applies infinitely often to remove a (σ_{true}, i) object ℓ. Between two consecutive stages $t_0 < t_1$ where Phase 4 is applied, we note that only (2.1) and (2.3) can be applied to remove ℓ. Both of these actions leave the target M-class unchanged. Hence the (σ_{true}, i) object formed by Phase 4 at t_0 and the (σ_{true}, i) object being removed at t_1 will both point at the same M-class. Since there are only finitely many objects ℓ' which are formed before stage s^*, this means that at almost every instance where Phase 4 applies to remove some ℓ, we have $M_\ell = M_{\ell'}$, where ℓ is a (σ_{true}, i) object and ℓ' is a (τ, j) object of higher priority formed after s^*. If infinitely many ℓ' are associated with a $\tau \subseteq \sigma_{true}$ then we apply Claim 3.25 to see that $i \in \widehat{\Xi}$. Since Phase 4 applies infinitely often we assume that infinitely many ℓ' are (τ, j) objects for some τ to the left of $\sigma_{true} * (Y + 1)$. Since ℓ' is formed after s^* we get a contradiction to Claim 3.24.

Case (2.1), (2.3). Now we assume that (2.4) and Phase 4 apply finitely often. Hence with finitely many exceptions every (σ_{true}, i) object is pointing at the same M-class M_y. Since $g_{\sigma_{true}}(y)$ is eventually stable, it is also impossible for either (2.1) or (2.3) to apply infinitely often.

This final contradiction shows that either (2.4) or Phase 4 must apply infinitely often, and hence $i \in \widehat{\Xi}$. This concludes the proof of Lemma 3.22. □

Finally we will demonstrate that $R = \{R_i\}_{i \in \omega}$ is Δ^0_2-categorical. Given any class R_i where $i > \min I_0^{\sigma_{true}}$ we use $\emptyset'$ to first check if $\#R_i > \max\{i, X_0\}$. Note that X_0 is a fixed constant with respect to i. If so we proceed to check if $i \in \widehat{\Xi}$ or if there is a stable σ_{true} link or clique involving i. At least one of the two alternatives is guaranteed to hold by Lemma 3.22. Note that $\widehat{\Xi}$ is a fixed c.e. set. If $i \in \widehat{\Xi}$ then $\#R_i = \infty$. Otherwise we apply the following:

Lemma 3.26. *If i is involved in a stable σ_{true} link or σ_{true} clique with pointer ℓ, then $\#R_i < \infty$ if and only if $g_{\sigma_{true}}(\ell) = f$.*

Proof. Since σ_{true} is the true node, $g_{\sigma_{true}}(\ell) = f$ iff $\#M_\ell < \infty$. Thus if $g_{\sigma_{true}}(\ell) = f$ then i must be involved in a stable σ_{true} link $\ell = \ell_{\sigma_{true}}(i)$. By Lemma 3.11 and the fact that σ_{true} is visited infinitely often, we see that $\#R_i < \infty$.

On the other hand if $g_{\sigma_{true}}(\ell) = \infty$ then $\#M_\ell = \infty$. By Lemma 3.10 we see that $\#R_i = \infty$. □

4. Categoricity of sets

Recall that an infinite Σ^0_2-set X is *categorical* if $E(X)$ is Δ^0_2-categorical. Fact 2.2 implies that $E(X)$ has a computable copy if and only if X is Σ^0_2. We follow Convention 2.5 and consider only infinite Σ^0_2 sets. By Proposition 2.6, a Σ^0_2 set X is categorical if and only if for every computable presentation of X there is some $g \leq_T \emptyset'$ telling the sizes of classes in this copy.

4.1. Comparing categoricity to other known properties

As we have seen, categorical sets are closed downwards under $\subseteq$ amongst Σ^0_2 sets. Since ω is limitwise monotonic (hence, is not categorical), categorical sets are not closed upwards under $\subseteq$, by Theorem 4.1 below. The second half of the theorem shows that, however, limitwise monotonicity fails to describe categorical sets in general. The simple result below is based on ideas contained in [7] and [21].

Theorem 4.1.

(1) *If an infinite Σ^0_2 set X is limitwise monotonic then X is not categorical.*
(2) *There exists an infinite Δ^0_2 set which is not categorical and not limitwise monotonic.*

Proof. (1) Recall that an infinite limitwise monotonic set is the range of some *injective* limitwise monotonic function. Suppose X is infinite and limitwise monotonic, and let f be an injective l.m.f. such that *range*$(f) = X$. By Proposition 2.6 it is sufficient to build a computable copy of $E(X)$ in which the size of the classes (the function $\#$) is not dominated by any $\emptyset'$-computable function. We use the limit lemma to fix an effective listing $(g_e)_{e\in\omega}$ of all approximations to partial $\emptyset'$ functions. We construct a copy M of $E(X)$ which satisfies the requirements:

$$R_y : \exists![x] \in M(\#[x] = f(y));$$
$$Q_e : g_e(3e)\downarrow \Rightarrow\ g_e(3e) < \#[3e] < [\infty].$$

There is also a global requirement which says $M \cong E(X)$.

The strategy for Q_e is to monitor $g_e(3e)$. If $g_e(3e)\downarrow$ and is greater or equal to the current size of the class $[3e]$, then we increase the size of $[3e]$ using a fresh z so that $f_s(z)$ is greater than the sizes of all equivalence classes introduced so far. More specifically, from this stage on we promise $\#[3e] = f(z)$.

The strategy for R_y introduces, if needed, a new class and declares its size to be $f(y)$. In the construction, if R_y is active for the first time, R_y picks a new fresh class of the form $[3j]$. At a later stage, R_y can be injured by Q_j. If this happens, R_y picks a new fresh $x = 3k + 1$ and declares $\#[3k + 1] = f(y)$.

In the construction, we build $E(X)$ by stages. We make $[3k + 2]$ infinite for every k, and we also let the strategies act according to their instructions. The verification is not difficult and is left to the reader.

(2) As in the proof of (1), we fix an effective listing $(g_e)_{e\in\omega}$ of approximations to partial Δ^0_2 functions. We construct a computable representation of an equivalence relation of the form $E(X)$ and satisfy:

$P_e : \lambda x \sup_z \varphi_e(x,z)$ is total and range $(\sup_z \varphi_e(x,z))$ is infinite $\Rightarrow (\exists y) \sup_z \varphi_e(y,z) \notin X$;
$Q_i : g_i$ is total $\to (\exists x)\, g_i(x) < \#[x] < \infty$;
$R_k : X$ contains at least k elements.

The strategy for P_e picks y such that $\sup_z \varphi_e(y,z)$, if it exists, is larger than 2^e (we will need more when we put the strategies together, see below). Notice that if *range* $(\sup_z \varphi_e(x,z))$ is infinite, then the strategy will eventually pick such an y. At stage s, the strategy keeps $\sup_z \varphi_e(y,z)[s]$ outside X increasing the size of $[x]$ to a larger finite value for every $x \leq s$ such that $\#[x]_s = \sup_z \varphi_e(y,z)[s]$.

The Q-strategy is similar to the one in the first part of the theorem. The witness does not have to be $3e$ and is picked by a strategy when it is initialized. The strategy for Q_i picks a fresh large x and makes $\#[x] > 2^e$. The strategy increases $\#[x]$ to be a larger number if necessary (we do not have a l.m.f. to find a safe spot, as in (1)).

The strategy for R_k introduces a class of size k, if this size is not restrained by P-strategies and is not currently among the sizes in the equivalence structure we are building.

The P and Q strategies have outcomes $\{\infty, \mathit{fin}\}$. We put the P and Q strategies onto a tree of strategies. In the construction, every strategy acts according to the outcomes of the strategies above it. If there is a P_e-strategy with outcome ∞ above a Q_i-strategy, then the Q_i-strategy waits for $\sup_z \varphi_e(x,z)$ to grow much larger than $g_{i,s}(z)$ (notice that Q_i may wait forever in the case when $g_i(z)$ tends to infinity or diverges, but this is fine). Every P-strategy will impose its restraint larger than the sizes of classes controlled by Q-strategies with finitary outcomes above it, and will impose its restraint to be less than the sizes of the classes controlled by higher priority Q-strategies above it having infinitary outcomes. □

Downey and Melnikov [12] showed that semi-lowness captures Δ^0_2-categoricity of completely decomposable groups. In the next result we use limitwise monotonicity and Theorem 4.1 to find an interesting relation between categorical sets and semi-low$_{1.5}$ sets. Recall that a set S is semi-low$_{1.5}$ if $\{x : W_x \cap S \text{ finite}\} \leq_1 \emptyset''$. We can equivalently replace $\emptyset''$ by $\mathit{Fin} = \{e : \mathrm{dom}(\varphi_e) \text{ finite}\}$.

Theorem 4.2.

(1) *Each infinite d.c.e. semi-low$_{1.5}$ set is not categorical.*
(2) *Some infinite superlow (hence semi-low$_{1.5}$) set is categorical.*

Comments on the proof. We prove the first part of the theorem by showing that each d.c.e. semi-low$_{1.5}$ set is limitwise monotonic, this fact is of an independent interest for us. The second half of the theorem is done by a direct construction which uses the usual lowness requirements.

Proof. (1). For the first half of the theorem, fix a d.c.e. semi-low$_{1.5}$ set S and a total computable p such that $W_e \cap S$ is finite if and only if $\mathrm{dom}(\varphi_{p(e)})$ is finite. We are building a limitwise monotonic function $f(x)$ using a c.e. set $W_{g(x)}$ whose index $g(x)$ is given by the recursion theorem. At stage s we will have $f_s(x)$, and then we will set $f(x) = \lim_s f_s(x)$.

Suppose $f_s(y)$ has already been defined for each $y < x$. In the following, we suppose that $f_s(y)$ has already reached their final values for every $y < x$ (we restart the procedure below, otherwise).

Consider the infinite d.c.e. set

$$S_x = \{v \in S : v > f_s(x-1)\} = U_x - V_x,$$

where U_x and V_x are some c.e. sets. Do the following:

(1) Start by enumerating all elements of U_x into $W_{g(x)}$ and wait for $\mathrm{dom}(\varphi_{pg(x)})$ to grow.
(2) As soon as $\mathrm{dom}(\varphi_{pg(x)})$ increases at a stage t, stop enumerating elements from U_x into $W_{g(x)}$ and set $f_t(x) = \min S_{x,t}$ which is the least $z \in U_{x,t}$ that has not been yet enumerated into $V_{x,t}$. (Without loss of generality, we may assume such a z exists, otherwise wait until it shows up.)
(3) If the current value of f enters V_x (thus leaves S permanently), pick the next largest z' currently in $W_{g(x),t}$ which has not yet entered V_x and set $f_{t'}(x) = z'$. Then repeat the same with the next largest z'' if z' leaves S, etc.
(4) If at some stage all elements from $W_{g(x),t}$ leave S, return to 1 above and repeat.

Notice that we cannot infinitely loop through (1) for in this case $W_{g(x)} \cap S$ is finite but $\mathrm{dom}(\varphi_{pg(x)})$ is infinite. Thus, there exists a stage s_0 and an element $c \in S$ such that $f_{s_0}(x)$ will be permanently set equal to c at stage s_0. Also, notice that $s \leq t$ implies $f_s(x) \leq f_t(x)$, and thus the function $f = \lim_s f_s$ is total and limitwise monotonic. Finally, the construction guarantees $f(x) < f(x+1)$, for every x, and therefore f is injective. (Note that a Σ^0_2-set that contains an infinite limitwise monotonic subset is limitwise monotonic, see e.g. [22].)

(2). Let $(Z_i)_{i\in\omega}$ be the effective listing of all partial computable models in the language of one binary predicate symbol. We are constructing an infinite Δ^0_2 set X so that the following requirements are met:

$L_e : \exists^\infty s \Phi^X_e(e)[s] \downarrow \Rightarrow \Phi^X_e(e) \downarrow$;
$R_j : Z_j$ represents E(X) $\Rightarrow \exists$ total $g_j \leq_T \emptyset'$ representing # in Z_j.

We split R_j further into sub-strategies, $R_{j,k}$:

$R_{j,k} : g_j$ guesses $\#[k]$ in Z_j correctly.

Remark: The construction will not be using a tree of strategies, for if we were using many versions of $R_{j,k}$, we would not be able to define g_j without an oracle for $\emptyset''$. In fact, it will be a finite injury construction. We also note that the lowness requirements L_e will ensure super-lowness if we can bound the number of injuries to each L_e by a computable function.

The strategy for $R_{j,k}$:

- Set a *threshold* for (the size of) $[k]$, a large and fresh number $\geq (\langle j,k\rangle+1)^2$ never seen in the construction before;
- At a stage t, keep $g_{j,t}(k) = \#_t[k]$ in Z_j unless $[k]$ passes its threshold, in the latter case set $g_{j,t}(k) = \infty$;
- If $[k]$ has passed its threshold at stage t, and currently $\#_t[k] \in X_t$, then extract x from X.

The strategy for L_e is a modification of the standard lowness strategy. More specifically, L_e attempts to preserve the computation $\Phi^X_e(e)[s]$ by restraining X on the use of $\Phi_e(e)[s]$. It can also put elements back to X, for the sake of restoring a computation of Φ_e which was previously seen but then was destroyed due to actions of higher priority R-substrategies. It does so unless this action injures higher priority strategies. Once the computation is restored, the strategy preserves that restored computation.

Construction. At stage 0, we set $X_0 = \omega$. At stage s, we let the strategies act according to their instructions.

Verification. By induction, we show that every L_e is met. In fact, we show that there exists a stage, after which $\Phi^X_e(e)[s] \downarrow$ implies $\Phi^X_e(e) \downarrow$. There are only finitely many R-substrategies that can potentially injure a computation of $\Phi^X_e(e)$. Suppose that after stage s all higher priority R-substrategies that correspond to finite witnesses are never active again, and suppose also that all higher priority L-strategies already passed their respective stages of stable evidence. If there is no $t \geq s$ at which $\Phi_e^X(e)[t] \downarrow$, then there is nothing to prove. Otherwise, suppose $\Phi_e^X(e)[t] \downarrow$ for $t \geq s$. There exists a stage $t' \geq t$ at which all higher priority R-strategies having infinitary behavior (i.e., having infinite classes as their witnesses) have their respective witnesses of sizes greater than the use of $\Phi_e^X(e)[t] \downarrow$. The strategy for L_e then restores the computation by returning missing elements into X. This computation will never be injured again. It is also easy to see that a bound for the number of injuries to each L_e can be computed in advance.

It is now straightforward to verify that $R_{e,j}$ is met, for every e, j. Since the strategy extracts elements from X, the respective structure Z_j must demonstrate it is isomorphic to $E(X)$ by growing the class. It is important that the substrategy can lift its threshold only finitely many times. Consequently, it eventually defines an astable threshold, and thus the whole process is Δ_2^0. Thus, $g_j \leq \emptyset'$, as desired. It is also clear that the set is infinite. □

Remark 4.3. It is not difficult to show that some d.c.e. set is categorical. We can modify the R-strategy so that whenever it extracts x from X it immediately puts $(x-1)$ back to X. The number x will never be put into X again. It is now sufficient to split ω into large enough intervals, and let "labels" move downwards within the intervals. We conclude that both conditions (being *d.c.e.* and being semi-low$_{1.5}$) are essential in Theorem 4.2(1).

Remark 4.4 *(Cholak).* In Theorem 4.2(1), semi-low$_{1.5}$ can be replaced by the "semilow$_2$ and the outer-splitting property", with essentially the same proof.

4.2. Degrees bounding categoricity

Although Theorem 4.1(2) implies that limitwise monotonicity fails to describe categorical sets, we would like to compare limitwise monotonic sets and categorical sets further. It is possible to describe c.e. degrees bounding infinite sets which are not limitwise monotonic:

Theorem 4.5. *(Downey, Kach, and Turetsky [14].) A c.e. degree $\boldsymbol{a}$ computes an infinite set which is not limitwise monotonic if and only if $\boldsymbol{a}$ is high.*

Since ω is limitwise monotonic, $S \oplus \omega$ is not categorical for a categorical set S. Thus, similarly to limitwise monotonicity, being categorical is not a degree-invariant property. Note that the property of being *not* categorical is, like being a limitwise monotonic set, closed upwards under $\subseteq$. There are more similarities of technical nature which occur when dealing with non-categorical sets. Our intuition is that non-categoricity is a non-uniform version of limitwise monotonicity. The intuition is: $E(X)$ is not Δ_2^0-categorical if (and only if) we can eventually provide each diagonalization substrategy with a sufficiently large class which will *monotonically* grow to a size $v \in X$. If X is limitwise monotonic, then it can be done with all uniformity and at once, but in general it does not have to be like that. This difference between non-categoricity and limitwise monotonicity is so subtle that c.e. degrees cannot distinguish them:

Theorem 4.6. *For a c.e. degree $\boldsymbol{a}$, the following are equivalent:*

(1) *$\boldsymbol{a}$ is high.*
(2) *There is some function $f \leq_T \boldsymbol{a}$ such that for every computable sequence of total computable functions $\{p_e\}$, there is a computable function g such that for each e, we have $f(x) > p_e(x)$ for every $x > g(e)$.*
(3) *There exists some infinite set $X \leq_T \boldsymbol{a}$ such that X is categorical.*

Proof. (3) ⇒ (1): Every infinite set computable from a non-high c.e. degree is limitwise monotonic and thus non-categorical, see Theorems 4.1 and 4.5.

(1) ⇒ (2): Let $\{p_e\}$ be a computable sequence of total computable functions. Let $P(x) = \sum_{e \leq x} p_e(x)$, where P is total computable. Then any dominant function computable from $\boldsymbol{a}$ must dominate P and hence p_e for every e. Fix (non-uniformly) a number x_0 such that $f(x) > P(x)$ for every $x > x_0$. Let $g(e) = \max\{x_0, e\}$.

(2) ⇒ (3): Fix a Turing functional Φ and a c.e. set A such that $f = \Phi^A$, where f satisfies (2). Fix an enumeration $\{A_s\}$ of A as well as an enumeration $\{C^i_n\}_{i,n\in\omega}$ of all uniformly c.e. sets. (Hence each computable equivalence relation is identified with some member of this sequence.) We may assume that at every stage s, $\Phi^A[s]$ converges on all inputs up to s.

We define an increasing sequence of markers $\{z_i\}$ by specifying an approximation $z_i[s]$ of z_i. We ensure that this approximation is increasing in i and s. Let B_i be the ith block, i.e., $B_i = [z_i, z_i + 2i^2]$, and $B_i[s]$ be the stage s approximation to B_i, i.e. $B_i[s] = [z_i[s], z_i[s] + 2i^2]$. Within the ith block we identify a unique element $x_i[s] \in B_i[s]$. At the end we take $X = \{\lim_s x_i[s] \mid i \in \omega\}$.

Construction of $\{z_i[s]\}$ *and* $\{x_i[s]\}$. To initialize B_i at stage s means to move z_i to a fresh number larger than s and beyond the boundaries of $B_0, \cdots, B_{i-1}$, and set $x_i = \max B_i$. At stage 0 initialize every B_i. Now assume we are at stage $s+1$. Let k be the least such that A has changed at stage $s+1$ below the use of $\Phi^A(k)[s]$ (if $A_{s+1} = A_s$ then we do nothing). Let i be the least such that $k \leq \max B_i$. We initialize B_j for every $j > i$. Suppose $k \leq x_i$ and there is some $i', j' < i$ such that $\#C^{i'}_{j'} = x_i$ currently. We decrease x_i by one, otherwise do nothing else in this stage. Now take $X = \{\lim_s x_i[s] \mid i \in \omega\}$.

It is easy to see that for every i and every s, the blocks $B_i[s]$ are pairwise disjoint and increasing in i. Furthermore each block is initialized finitely often and A can compute a stage where each z_i and B_i are stable. Each x_i must stay within the block B_i, because it initially starts off as $\max B_i$, and is decreased each time we find some $\#C^{i'}_{j'} = x_i$, where $i', j' < i$. Since the size of each class $\#C^{i'}_{j'}$ is non-decreasing, and the size of the block B_i is $2i^2 + 1$, x_i will never leave the block. Since each initialization to a block moves it to a fresh location, it is easy to see that $X \leq_T A$ (knowing the function f allows us to compute where the blocks are), and that X is infinite. Let $X_s = \{x_i[s] \mid i < s\}$.

We now claim that X is Δ^0_2-categorical. Fix $\{C_n\} = \{C^I_n\}$ and assume that $\{C_n\}$ is an equivalence structure presenting $E(X)$. Define $\{p_n\}$ by the following. Run the approximation for $\{X_s\}$ and $C_n[s]$, and suppose we have defined $p_n(x)$ at stage s_x. We search for a stage $s_{x+1} > s_x$ such that either (i) $\#C_n[s_{x+1}] \in X_{s_{x+1}}$, or (ii) $X_{s_{x+1}} \cap (\#C_n[s_x], \#C_n[s_{x+1}]] \neq \emptyset$. When s_{x+1} is found we define $p_n(x+1)$ to be larger than the current value of $f(x+1)$.

Claim 4.7. *p_n is total for every n.*

Proof. If p_n is not total then there is some least s_{x+1} which we fail to find. Since (i) does not hold after stage s_x, we can conclude that C_n is infinite (C_n cannot be finite because $\{C_k\}$ is assumed to be an equivalence structure presenting $E(X)$). Since X is infinite we must have (ii) holds at some large stage after s_x, a contradiction. □

Now fix a computable function g such that for each n, we have $f(x) > p_n(x)$ for every $x > g(n)$. Now fix n. Let t be first stage such that $\#C_n[t] > \max\{n, I, g(n)\}$, and such that B_{i+1} is initialized at stage t where i is the largest such that $\max B_i < \#C_n[t]$ at stage t. If t exists we define $h(n) = \infty$, otherwise we define $h(n) = f$. Clearly $h \leq_T \emptyset'$; in fact there is a computable approximation to $h(n)$ which changes at most once on each input n.

If t does not exist, we argue that C_n is finite. Otherwise for almost every i, at the first stage where $\#C_n$ grows larger than $\max B_i$, we can conclude that A is stable below the use for $\Phi^A \upharpoonright \max B_i + 1$. This allows us to compute A, which is impossible since A is high.

Now finally assume that t exists. We argue that C_n is infinite. Let $s_{x_0} > t$ be the least stage of this form. We claim that there are infinitely many $x > x_0$ such that $\#C_n[s_x] \geq x$. Note that $\#C_n[s_{x_0}] \geq x_0$, because at stage t we would have ensured that the interval $[\#C_n[t], t] \cap X_t = \emptyset$, and by construction of X we in fact have $[\#C_n[t], t] \cap X_{t'} = \emptyset$ for every $t' \geq t$. Since $p_n(x_0 - 1)[t] \downarrow$ we have that $x_0 \leq t$. Clearly at stage s_{x_0} we must have $\#C_n[s_{x_0}] > t \geq x_0$.

Now suppose that there are only finitely many $x \geq x_0$ such that $\#C_n[s_x] \geq x$. Since x_0 is such a stage, we assume that x is the largest such that $\#C_n[s_x] \geq x$. By maximality of x we have $\#C_n[s_x] = x$, and in fact we must have $\#C_n[s_{x+1}] = x$. We have $x = \#C_n[s_x] \geq \#C_n[t] > g(n)$. Since f dominates p_n and $p_n(x) > \Phi^A(x)[s_x]$, this means that A has to change below the use of $\Phi^A \restriction x + 1$ after stage s_x.

Claim 4.8. *At stage s_x, there is some j such that $x \in B_j[s_x]$, where we have $x_j[s_x] \leq x$.*

Proof. Suppose that x is not in any block. Then (ii) must hold at stage s_x. Let j' be the largest block such that $\max B_{j'} < x$. Obviously $\max B_{j'} > \#C_n[s_{x-1}]$. Clearly at the previous stage s_{x-1} the block $B_{j'}$ was still in the same position, and thus $x_{j'}[s_{x-1}] \geq x_{j'}[s_x]$, which would contradict the choice of s_x. Hence there is some j such that $x \in B_j[s_x]$.

At stage s_x if (i) holds then $\#C_n[s_x] \in X_{s_x}$ and the claim certainly holds. Otherwise (ii) holds which means we have a new element $x_{j'}[s_x]$ such that $C_n[s_{x-1}] < x_{j'}[s_x] \leq C_n[s_x]$. Clearly $j' = j$ because otherwise $j' < j$ and a contradiction can be derived as above. □

Now by Claim 4.8 we can conclude that when A next changes, say at stage $u > s_x$, below the use of $\Phi^A \restriction x + 1$, we must have the interval $[\#C_n[u], u] \cap X_u = \emptyset$. In fact, by the construction we have $[\#C_n[u], u] \cap X = \emptyset$. Let $y > x$ be the least such that $s_y > u$; since $s_{y-1} \leq u$, in particular we have $y \leq u$. Furthermore at stage s_y we must have $\#C_n[s_y] \geq u \geq y$, since the whole interval $[\#C_n[u], u]$ is disjoint from X. This contradicts the maximality of x. □

5. A short conclusion

We leave open the following:

Question 5.1. *(See [7].) Which computable equivalence structures are Δ^0_2-categorical?*

It may very well happen that no classical notion of computability theory (nor any reasonable combination of such properties) captures categoricity of a set. In this case we would like to know more about such sets.

We hope that our techniques can be used to attack the following problem:

Question 5.2. *Describe Δ^0_2-categorical linear orders.*

We would also like to know more about Δ^0_2-degrees of categoricity of computable equivalence structures:

Question 5.3 *(Csima). Is every Δ^0_2 degree of categoricity of a computable equivalence structure either complete or computable?*

References

[1] U. Andrews, S. Lempp, J. Miller, K.M. Ng, L. San Mauro, A. Sorbi, Universal computably enumerable equivalence relations, J. Symbolic Logic 79 (1) (2014) 60–88.

[2] C.J. Ash, Recursive labeling systems and stability of recursive structures in hyperarithmetical degrees, Trans. Amer. Math. Soc. 298 (1986) 497–514.

[3] C.J. Ash, J. Knight, Computable Structures and the Hyperarithmetical Hierarchy, Studies in Logic and the Foundations of Mathematics, vol. 144, North-Holland Publishing Co., Amsterdam, 2000.

[4] Ewan J. Barker, Back and forth relations for reduced abelian p-groups, Ann. Pure Appl. Logic 75 (3) (1995) 223–249.

[5] N. Bazhenov, δ_2^0-Categoricity of boolean algebras, Vestnik, Quart. J. Novosibirsk State Univ. 13 (2) (2013) 2–14.

[6] Howard Becker, Alexander S. Kechris, The Descriptive Set Theory of Polish Group Actions, London Mathematical Society Lecture Note Series, vol. 232, Cambridge University Press, Cambridge, 1996.

[7] Wesley Calvert, Douglas Cenzer, Valentina Harizanov, Andrei Morozov, Effective categoricity of equivalence structures, Ann. Pure Appl. Logic 141 (1–2) (2006) 61–78.

[8] J. Carson, V. Harizanov, J. Knight, K. Lange, C. McCoy, A. Morozov, S. Quinn, C. Safranski, J. Wallbaum, Describing free groups, Trans. Amer. Math. Soc. 364 (11) (2012) 5715–5728.

[9] Samuel Coskey, Joel David Hamkins, Russell Miller, The hierarchy of equivalence relations on the natural numbers under computable reducibility, Computability 1 (1) (2012) 15–38.

[10] Barbara F. Csima, Johanna N.Y. Franklin, Richard A. Shore, Degrees of categoricity and the hyperarithmetic hierarchy, Notre Dame J. Form. Log. 54 (2) (2013) 215–231.

[11] Denis, R. Hirschfeldt, Bakhadyr Khoussainov, Richard A. Shore, Arkadii M. Slinko, Degree spectra and computable dimensions in algebraic structures, Ann. Pure Appl. Logic 115 (1–3) (2002) 71–113.

[12] Rodney Downey, Alexander G. Melnikov, Effectively categorical abelian groups, J. Algebra 373 (2013) 223–248.

[13] Rodney Downey, Alexander G. Melnikov, Computable completely decomposable groups, Trans. Amer. Math. Soc. 366 (8) (2014) 4243–4266.

[14] R. Downey, A. Kach, D. Turetsky, Limitwise monotonic functions and applications, in: Proceedings of the Eleventh Asian Logic Conference, 2011, pp. 59–85.

[15] R. Downey, A. Melnikov, K.M. Ng, Abelian groups and the halting problem, Preprint.

[16] Yu.L. Ershov, Problems of Solubility and Constructive Models, Nauka, Moscow, 1980 [in Russian].

[17] Yuri L. Ershov, Sergei S. Goncharov, Constructive models, in: Siberian School of Algebra and Logic, Consultants Bureau, New York, 2000.

[18] Ekaterina B. Fokina, Sy-David Friedman, Asger Törnquist, The effective theory of Borel equivalence relations, Ann. Pure Appl. Logic 161 (7) (2010) 837–850.

[19] E. Fokina, J.F. Knight, A. Melnikov, S.M. Quinn, C. Safranski, Classes of Ulm type and coding rank-homogeneous trees in other structures, J. Symbolic Logic 76 (3) (2011) 846–869.

[20] E. Fokina, A. Frolov, I. Kalimullin, Categoricity spectra for rigid structures, Notre Dame J. Form. Log. (2015), in press.

[21] Asher M. Kach, Daniel Turetsky, Δ_2^0-Categoricity of equivalence structures, New Zealand J. Math. 39 (2009) 143–149.

[22] Iskander Kalimullin, Bakhadyr Khoussainov, Alexander Melnikov, Limitwise monotonic sequences and degree spectra of structures, Proc. Amer. Math. Soc. 141 (9) (2013) 3275–3289.

[23] N.G. Khisamiev, Constructive abelian groups, in: Handbook of Recursive Mathematics, Vol. 2, in: Studies in Logic and the Foundations of Mathematics, vol. 139, North-Holland, Amsterdam, 1998, pp. 1177–1231.

[24] Bakhadyr Khoussainov, Andre Nies, Richard A. Shore, Computable models of theories with few models, Notre Dame J. Form. Log. 38 (2) (1997) 165–178.

[25] A.H. Lachlan, A note on positive equivalence relations, Z. Math. Log. Grundl. Math. 33 (1) (1987) 43–46.

[26] Karen Lange, Robert I. Soare, Computability of homogeneous models, Notre Dame J. Form. Log. 48 (1) (2007) 143–170 (electronic).

[27] P. LaRoche, Recursively presented boolean algebras, Notices Amer. Math. Soc. 24 (1977) 552–553.

[28] Ch.F.D. Mak-koĭ, On Δ_3^0-categoricity for linear orders and Boolean algebras, Algebra Logika 41 (5) (2002) 531–552, 633.

[29] Charles F.D. McCoy, Δ_2^0-Categoricity in Boolean algebras and linear orderings, Ann. Pure Appl. Logic 119 (1–3) (2003) 85–120.

[30] Charles McCoy, John Wallbaum, Describing free groups, Part II: Π_4^0 hardness and no Σ_2^0 basis, Trans. Amer. Math. Soc. 364 (11) (2012) 5729–5734.

[31] A.T. Nurtazin, Computable Classes and Algebraic Criteria of Autostability, Summary of Scientific Schools, Math. Inst. SB USSRAS, Novosibirsk, 1974.

[32] J.B. Remmel, Recursively categorical linear orderings, Proc. Amer. Math. Soc. 83 (2) (1981) 387–391.

[33] Robert I. Soare, Recursively Enumerable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets, Perspectives in Mathematical Logic, Springer-Verlag, Berlin, 1987.