Degrees of Categoricity and the Isomorphism Problem

by

Mohammad Assem Abd-Alqader Mahmoud

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Pure Mathematics

Waterloo, Ontario, Canada, 2019

Examining Committee Mambership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	NAME: Douglas Cenzer Title: Professor
Supervisor(s)	NAME: Barbara F. Csima Title: Professor
Internal Member	NAME: Ross Willard Title: Professor
Internal-external Member	NAME: Bruce Richter Title: Professor
Other Member(s)	NAME: Rahim Moosa Title: Professor

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

I hereby declare that I am the sole author of this text, with exception of Chapter 3. Chapter 3 is a modified version of a joint work with Csima, Deveau and Harrison-Trainor [12].

Abstract

In this thesis, we study notions of complexity related to computable structures. We first study degrees of categoricity for computable tree structures. We show that, for any computable ordinal α, there exists a computable tree of rank $\alpha+1$ with strong degree of categoricity $\mathbf{0}^{(2\alpha)}$ if α is finite, and with strong degree of categoricity $\mathbf{0}^{(2\alpha+1)}$ if α is infinite. For a computable limit ordinal α, we show that there is a computable tree of rank α with strong degree of categoricity $\mathbf{0}^{(\alpha)}$ (which equals $\mathbf{0}^{(2\alpha)}$).

In general, it is not the case that every Turing degree is the degree of categoricity of some structure. However, it is known that every degree that is of a computably enumerable (c.e.) set in and above $\mathbf{0}^{(\alpha)}$, for α a successor ordinal, is a degree of categoricity. In this thesis, we include joint work with Csima, Deveau and Harrison-Trainor which shows that every degree c.e. in and above $\mathbf{0}^{(\alpha)}$, for α a limit ordinal, is a degree of categoricity. We also show that every degree c.e. in and above $\mathbf{0}^{(\omega)}$ is the degree of categoricity of a prime model, making progress towards a question of Bazhenov and Marchuk.

After that, we study the isomorphism problem for tree structures. It follows from our proofs regarding the degrees of categoricity for these structures that, for every computable ordinal $\alpha > 0$, the isomorphism problem for trees of rank α is $\Pi_{2\alpha}$-complete. We also discuss the isomorphism problem for pregeometries in which dependent elements are dense and the closure operator is relatively intrinsically computably enumerable. We show that, if K is a class of such pregeometries, then the isomorphism problem for the class K is Π_3-hard.

Finally, we study the Turing ordinal. We observed that the definition of the Turing ordinal has two parts each of which alone can define a specific ordinal which we call the upper and lower Turing ordinals. The Turing ordinal exists if and only if these two ordinals exist and are equal. We give examples of classes of computable structures such that the upper Turing ordinal is β and the lower Turing ordinal is α for all computable ordinals $\alpha < \beta$.

Acknowledgements

To my sui generis father, the master and commander, Assem Hamdy: Thank you for architecting my life for the past three decades.

I wish that my mother Nazek Ayoub could witness this finally happening. I am dedicating this thesis to her because I believe she is the one who endured the most to help me grow. I also wish that I could tell my grandmother Samiha Hemmat about this.

To my siblings Amena and Ali: I am grateful for having you as informative colleagues and reliable family members.

To my supervisor Barbara Csima: I am thankful for your kindness, patience, and keeping things well on track for the past half of a decade.

I would like to thank the department officers Nico Spronk and David McKinnon; and to thank the staff Jackie Hilts, Lis D'Alessio, Nancy Maloney, and Pavlina Penk for their unfailing planning and assistance.

I appreciate my examining committee: Douglas Cenzer, Bruce Richter, Raheem Moosa, and Ross Willard for their careful reading and for their valuable feedback.

It was a great opportunity working with Michael Deveau and Matthew Harrison-Trainor, as well as the many other great colleagues. Special thanks to Suzanne Findleton, Stanley Xiao, and the most recent José Luis Avilez.

I would have not made it to this point without the help of some of the people I knew during my time at Cairo University. Thanks to Tarek Sayed Ahmed, Ahmed El-Guindy, Gonzalo Aranda Pino, Gábor Sági, Hany Elhosseiny, Nefertiti Megahed, Eslam Essam, and Diaaeldin Taha.

I finally want to thank the salsa commuinty, the Society of Arab Students, and all the amazing people I met.

Thank you God for all I have.

Table of Contents

Chapter 1

Introduction

Computability theorists have developed techniques for studying sets of natural numbers. These techniques can be applied more generally for studying other mathematical objects that can be coded into natural numbers. Many structures like linear orders, graphs, trees and algebraic structures, by Gödel numbering their atomic diagrams, can be coded into sets of natural numbers. Hence, one can talk about the complexity of such structures, their copies and the isomorphisms between pairs of these copies from a computability theoretic perspective.

As mentioned in the abstract, we study here three notions of complexity related to structures and their isomorphisms. After the introduction (Chapter 1), we study degrees of categoricity (Chapters 2,3), the isomorphism problem (Chapter 4) then we study the Turing ordinal (Chapter 5). The work in this thesis resulted into three papers ([28], [12], [29]).

1.1 Basic Computability Review

Turing Programs and Partial Computable Functions

In computability theory we study the idea of using an effective procedure (algorithm) for answering mathematical questions. Informally, by an effective procedure we mean a set of step-by-step unambiguous instructions (think of a computer program). Formally, we mean a Turing Machine (see [36] for a def-

inition). We believe that a function is effectively calculable by a human being if and only if it can be computed by a Turing machine. That belief is known as *Turing's Thesis*.

When we talk about a set of instructions as in a program, or an algorithm, we are talking about a finite sequences of lines where each line is a sequence of symbols from a finite alphabet. In a finite alphabet, one can always effectively code its characters by natural numbers. This implies that a program is just a finite sequence of natural numbers; hence, a program can always be coded as a single natural number (Gödel number). Accordingly, we can list all Turing programs and give them indices in such a way that for any program we can effectively find its index, and conversely, from the index we can find the program. This is called the *standard numbering* or *canonical numbering* of Turing programs.

Let P_e denote the Turing program with Gödel number e. We write $\varphi_e(x) = y$ if program P_e with input x halts with output y. For every natural number e, we can view φ_e as a partial function on the set of natural numbers; hence, φ_e is called a *partial computable (p.c.) function*. When the domain of φ_e is all the natural numbers, we remove the word "partial". In other words, a function $f : \omega \to \omega$ is *computable* if there is a natural number e such that $f = \varphi_e$, and φ_e is total (with domain ω).

Fact (The Padding Lemma): Each p.c. function has infinitely many indices (see [36]).

Oracle Turing Programs and Relative Computability

Sometimes we answer a question 'Q' using the knowledge of the answer to another question 'P' without worrying about how the answer to 'P'is established. This motivates the idea of an *Oracle Turing Machine*; see [36] for definition. Informally, if A is a set of natural numbers, a program with oracle A is one that allows the following to be one of its step-by-step instructions: *If* $x \in A$, *then ... , else, ...*

For two sets of natural numbers X and Y, we say that X is *Turing reducible to* Y (or, that X is computable relative to Y) if there is a Turing pro-

gram, with Y as an oracle, which answers the question: "Does $x \in X$?" for every natural number x. In symbols, we write $X \leq_T Y$. We also write $X <_T Y$ to mean that $X \leq_T Y$ but not $Y \leq_T X$.

Let $\{\tilde{P}_e\}_{e\in\omega}$ be an effective numbering of the Turing oracle programs. We write Φ_e^A to mean the p.c. function that can be computed by $\tilde{P}_e$ using A as an oracle. We regard Φ_e as a (partial) functional on sets (from 2^ω to 2^ω) mapping A to B if $B = \Phi_e^A$ (we identify sets with their characteristic functions). This means that, for two sets of natural numbers X and Y, the set X is *Turing reducible to* Y if, for some natural number e, $X = \Phi_e^Y$.

Remark: The fact that programs are finite implies that, if a computation of the program $\tilde{P}_e$ with oracle A halts, then only finitely many bits of A were used in that computation (see Theorem 3.3.9. in [36]).

Multi-variable P.C. Functions and the Lambda Notation

We defined p.c. functions with a single argument. We will allow writing several arguments. This is safe because one can effectively transition from several arguments to a single argument using the *standard pairing function* (see [36]). The pairing function has two arguments, and is denoted by brackets $\langle , \rangle$ (like inner product notation). For $n = \langle x, y\rangle$, we let $\pi_1(n) = x$ and $\pi_2(n) = y$ (i.e. π_1 and π_2 are the projections).

The situations in which we have several arguments require us, sometimes, to distinguish which are variables and which are parameters. The Church *lambda notation* for partial functions helps in such situations. Suppose that $[\ldots x \ldots]$ is a mathematical expression such that, for every integer x, $[\ldots x \ldots]$ has at most one corresponding value. Then, $\lambda x[\ldots x \ldots]$ denotes the corresponding partial function $x \mapsto [\ldots x \ldots]$ (e.g. $\lambda x[x^2]$). We also use the lambda notation for partial functions of multiple variables; $\lambda xy[x + y]$, say. Note now the difference between $\lambda xy[x + y]$ and $\lambda x[x + y]$. The first denotes addition as a function of x and y. However, the second indicates that the expression is viewed as a function of x with y as a parameter.

We will not use the lambda notation in this thesis except for the statement of the next theorem.

The *s-m-n* Theorem

The following is a fundamental result that allows us to move arguments around effectively. For a proof, see [36].

Theorem 1.1.1 (The (Relativized) *s-m-n* Theorem)**.** *For every $m, n \geq 1$, there exists an injective computable function s_n^m of $m+1$ variables such that for all sets $A \subseteq \omega$ and for all $x, y_1, \ldots, y_m \in \omega$,*

$$\Phi^A_{s_n^m(x,y_1,\ldots,y_m)} = \lambda z_1, \ldots, z_n[\Phi^A_x(y_1, \ldots, y_m, z_1, \ldots, z_n)].$$

The last theorem is also known as the Parameter Theorem.

Computable Approximations

We write $\varphi_{e,s}(x) = y$ if $x, y, e < s$ and y is the output of $\varphi_e(x)$ in $< s$ steps of the Turing program P_e. When there is an oracle A, and if the use of the computation is $< s$, we write $\Phi^A_{e,s}(x) = y$.

Computably Enumerable (C.E.) Sets

A set is computably enumerable (c.e.) if it is the domain of some a p.c. function. We define the e^{th} c.e. set to be $W_e := dom(\varphi_e)$. A c.e. set W_e can be approximated (see Definition1.1.3 below) by the finite sets $W_{e,s} := dom(\varphi_{e,s})$. Similarly, we have the e^{th} c.e. in X set $W_e^X := dom(\Phi_e^X)$. We also have the approximations $W^X_{e,s} = dom(\Phi^X_{e,s})$.

The Arithmetical Hierarchy

We have just defined what it means to be a c.e. set. Note that, $x \in W_e$ iff $\exists s(x \in W_{e,s})$. If we regard $x \in W_{e,s}$ as a relation $R(x, s)$ on $\omega \times \omega$, then R is computable. A set A is said to be Σ^0_1 if it is of the form $\{x : (\exists y)R(x, y)\}$. This means that c.e. sets are Σ^0_1. In fact, we can show that every Σ^0_1 set is c.e. (see Theorem 2.1.3. in [36]). This makes Σ^0_1 exactly the class of c.e. sets.

Now we define the more general notion of Σ^0_n sets for every natural number n. The superscript 0 is used to indicate that the sets within the class Σ^0_n are

definable by formulas that use quantification on numbers and that no quantification is on functions or sets. Since we will never use function quantifiers in this thesis, we will drop the superscript 0.

Definition 1.1.2. (1) A set is Σ_0 (also Π_0, Δ_0) iff it is computable.

(2) We defined Σ_1 earlier. The class of complements of Σ_1 sets is denoted by Π_1. Define $\Delta_1 := \Sigma_1 \cap \Pi_1$; this is exactly the computable sets.

(3) For $n > 1$, a set is Σ_n if there is a Π_{n-1} relation R such that $x \in A \Leftrightarrow (\exists y) R(x, y)$. The class of complements of Σ_n sets s denoted by Π_n. Define $\Delta_n := \Sigma_n \cap \Pi_n$.

A set is arithmetical if it belongs to $\bigcup_n (\Sigma_n \cup \Pi_n)$.

We can relativize the arithmetical hierarchy with respect to an oracle A by replacing the word computable with A-computable in the definition. We use the notation Σ_n^A, Π_n^A, and Δ_n^A to denote the relativized hierarchy.

One can show that (see Theorem 4.1.4 in [36]):

(1) if a set A is Σ_n (or Π_n), then $A \in \Delta_m$ for every $m > n$;

(2) the classes Σ_n, Π_n, and Δ_n are closed under $\cap$ and $\cup$.

Noncomputable C.E. Sets and the Turing Jump

Consider the following set $K := \{e : \varphi_e(e) \text{ halts } \} = \{e : e \in W_e\}$. It is easy to see that K is domain of a p.c. function; namely, $\psi(x) = x$ if $\varphi_x(x)$ halts, and undefined otherwise. This means that K is computably enumerable. One can show that K is not computable by a standard diagonalization argument. Indeed, if K was computable, then one can bulid the following computable function:

$$f(x) = \begin{cases} \varphi_x(x) + 1 & \text{if } x \in K \\ 0 & \text{if } x \notin K \end{cases}$$

Clearly, $f \neq \varphi_x$ for every x; hence, a contradiction.

With the development of relative computability, Kleene and Post defined K^A which is a relativization of the definition of K to an oracle A; $K^A := \{e :$

$e \in W_e^A\}$. They also noted that, by relativizing the argument above, $K^A >_T A$ and K^A is c.e. in A. The set K^A is known as the *jump* of A and is denoted A'. The prime sign is known as the *jump operator*. One can iterate the jump n times (finite n); we define $A^{(0)} = A$, $A^{(n+1)} = (A^{(n)})'$. Note that $A^{(1)} = A'$.

Turing Degrees

For two sets A, B of natural numbers, we write $A \equiv_T B$ if $A \leq_T B$ and $B \leq_T A$. It is not hard to see that $\equiv_T$ is reflexive, transitive, and symmetric which means that $\equiv_T$ is an equivalence relation. The *Turing degree* of a set A is the equivalence class $\mathbf{deg}(A) = \{B : B \equiv_T A\}$. We can define a partial ordering $\leq$ on the set of Turing degrees as follows: $\mathbf{deg}(A) \leq \mathbf{deg}(B)$ iff $A \leq_T B$. We write $\mathbf{deg}(A) < \mathbf{deg}(B)$ if $A <_T B$, i.e., if $A \leq_T B$ and $B \not\leq_T A$.

We mostly use boldface small letters to denote Turing degrees (e.g. $\mathbf{a}, \mathbf{b}, \ldots$). We use $\mathbf{0}$ (bold-face 0) to denote the Turing degree of the empty set (the computable degree). We will just say a *degree* to mean a Turing degree.

It is helpful to define the Turing jump for degrees in the following, obviously well defined, way: $(\mathbf{deg}(A))^{(n)} = \mathbf{deg}(A^{(n)})$.

A degree $\mathbf{a}$ is *computably enumerable* if it contains a c.e. set. A degree $\mathbf{a}$ is *computably enumerable in* another degree $\mathbf{b}$ if $\mathbf{a}$ contains some set A which is c.e. in some set $B \in \mathbf{b}$. A degree $\mathbf{a}$ is *computably enumerable in and above* $\mathbf{b}$ if $\mathbf{a}$ is c.e. in $\mathbf{b}$ and $\mathbf{a} \geq \mathbf{b}$.

Uniform Computability and Limit Computable Sets

We mentioned earlier how c.e. sets can be approximated by a sequence of finite sets. Note that the members of such a sequence are generated computably (we can effectively, given s, compute $W_{e,s}$). We can more generally talk about approximations by sequences of computable sets. It makes sense to also assume that, not only every member of the sequence is a computable set, but also there is a computable procedure with which we can generate those members. Such are called *uniformly computable sequences*. More precisely, a sequence of computable functions $\{f_s(x)\}_{s\in\omega}$ (or sets) is *uniformly computable* if there is a computable function $g(x, s) = f_s(x)$ for all natural s and x.

Now consider the following natural definition:

Definition 1.1.3. [See [36]] A set A is *limit computable* if there is a uniformly computable sequence $\{A_s\}_{s\in\omega}$ such that for all x,

$$A(x) = \lim_s A_s(x).$$

An associated concept with limit computability is that of a *modulus*. Given A and $\{A_s\}_{s\in\omega}$ such that $A = \lim_s A_s$, a function $m(x)$ is a *modulus (of convergence)* if

$$(\forall x)(\forall s \geq m(x))(\forall y \leq x)[A(y) = A_s(y)].$$

Note that $A \leq_T m$.

The following is a key result which we will rely on in Chapter 4 (for a proof, see Theorem 3.6.5 in [36]).

Theorem 1.1.4 (Modulus Criterion)**.** *A set A has a c.e. degree iff $A = \lim_s A_s$ for a computable sequence $\{A_s\}_{s\in\omega}$ with modulus $m(x) \leq_T A$.*

Computable Ordinals, Transfinite Jumps, and the Hyperarithmetical Hierarchy

We saw how one defines the iterated jump $A^{(n)}$ and the arithmetical hierarchy. Both are defined inductively. Such inductive definitions can be carried transfinitely along higher ordinals (e.g. $A^{(\omega)}, A^{(\omega+1)}, \ldots$). However, we cannot effectively inductively go beyond the countable ordinals. In fact, the class of countable ordinals is too big (is itself uncountable); hence, we will be only confined to those countable ordinals which we can effectively code by natural numbers. The latter class of ordinals is called the *computable ordinals*. Rigorously, an ordinal is computable if it is a well ordering $(X, <)$ where X is a computable set and $<$ is a computable binary relation on X (recall that ordinals are just well orderings, and that every well ordering is isomorphic to an ordinal). One can show that computable ordinals form an initial segment

of the ordinals (see Proposition 4.6 in [4]). It is not hard to see that the first non-computable ordinal is strictly less than ω_1.

In what follows we describe Kleene's system of notations that codes computable ordinals as natural numbers. We define simultaneously a set of notations $\mathcal{O}$, a function $|\ |_{\mathcal{O}}$ from $\mathcal{O}$ to the class of computable ordinals, and a strict partial ordering $<_{\mathcal{O}}$ on $\mathcal{O}$.

(1) Let $|1|_{\mathcal{O}} = 0$ (i.e. 1 is the notation for 0).

(2) If $|a|_{\mathcal{O}} = \alpha$, then $|2^a|_{\mathcal{O}} = \alpha + 1$; define $b <_{\mathcal{O}} 2^a$ iff $b = a$ or $b <_{\mathcal{O}} a$.

(3) For a limit ordinal α, $\alpha = |3.5^e|_{\mathcal{O}}$ for every e such φ_e is total computable, with values in $\mathcal{O}$, where

$$\varphi_e(0) <_{\mathcal{O}} \varphi_e(1) <_{\mathcal{O}} \varphi_e(2) <_{\mathcal{O}} \dots,$$

and α is the least upper bound of the sequence of ordinals $\alpha_n = |\varphi_e(n)|_{\mathcal{O}}$. Define $b <_{\mathcal{O}} 3.5^e$ if there exists n such that $b <_{\mathcal{O}} \varphi_e(n)$.

This finishes describing the system of notations. The class of ordinals having notations in $\mathcal{O}$ is called *constructive ordinals*. For an elaborate discussion of why constructive ordinals are exactly the computable ones, see [4].

Note that, in general, it is not the case that every ordinal has a unique notation in $\mathcal{O}$. In fact, only finite ordinals do. This is because notations for limit ordinals are defined as the index of some computable sequence of notations which we call a *fundamental sequence*. In practice, we a pick large enough limit ordinal below which we restrict ourselves to a subset of the notations so that every limit ordinal has a unique fundamental sequence which we require to contain only successor ordinals and to start with 1.

Now, given a set A, we can extend the sequence of jumps $A, A', A'', \dots$ effectively transfinitely. For a computable limit ordinal α (e.g. $\alpha = \omega$), let $A^{(\alpha)} = \{\langle u, v\rangle : u <_{\mathcal{O}} 3.5^e \text{ and } v \in A^{(u)}\}$, where $\alpha = |3.5^e|_{\mathcal{O}}$. For every computable ordinal β, $A^{(\beta+1)} = (A^{(\beta)})'$. Rigorously, the transfinite recursion we have just described is done in terms of the notations from $\mathcal{O}$ (see [4]).

We know what are Σ_n, Π_n, and Δ_n for every finite n. At this point we can expand the definitions to any computable ordinal α instead of n. For a

computable ordinal α, a set A is Σ_α if it is definable by a formula of the form $\bigvee_{k\in\omega} \exists \bar{y}_k R_k(x, \bar{y}_k)$ where $\{R_k(x, \bar{y}_k)\}_{k\in\omega}$ is a c.e. set of formulas such that, for every k, $R_k(x, \bar{y}_k)$ is Π_{β_k}, $\beta_k < \alpha$. Accordingly, $\Sigma_\alpha = \Sigma_1^{\varnothing^{(\alpha)}}$ for every limit ordinal α.

A set is Π_α if its complement is Σ_α. A set is Δ_α if it is both Σ_α and Π_α.

Many-one Reducibility and Complete Sets

Within a class of sets, it is useful sometimes to know sets that are the, in some sense, the hardest in that class.

A set A is said to be *m-reducible (or many-one reducible)* to a set B if for some computable function f, for every $x \in \omega$, $x \in A$ iff $f(x) \in B$.

For a class of sets Γ, a set C is Γ-complete if

(1) $C \in \Gamma$,

(2) C is Γ-hard; that is, every set $X \in \Gamma$ is m-reducible to C.

The following are examples of complete sets (for proofs, see Theorem 4.2.2. in [36] for the finite case; the infinite case is just a relativization).

(1) $\varnothing^{(n)}$ is Σ_n-complete for every finite n.

(2) $\varnothing^{(\alpha+1)}$ is Σ_α-complete for every computable infinite α.

1.2 Computable Structure Theory Review

1.2.1 Degrees of Categoricity

We mentioned earlier that one can talk about mathematical structures that can be coded into natural numbers. From a computably theoretic perspective, our main objects will be *computable structures*.

Definition 1.2.1. [Computable Structure [4]] A structure is computable if it has a computable atomic diagram and a computable universe.

In most of mathematics we consider isomorphic structures as the same thing; however, in a world in which structures are computable, two isomorphic structures may have different computational properties. This happens when there are only non-computable isomorphisms between the two computable copies. Structures for which this issue does not show up we call *computably categorical* (see [4]); more precisely, a computable structure $\mathcal{C}$ is *computably categorical* if, for every computable $\mathcal{A}$ such that $\mathcal{A} \cong \mathcal{C}$, there is a computable isomorphism from $\mathcal{C}$ to $\mathcal{A}$.

One can generalize the definition to study situations when the isomorphisms are not necessarily computable but computable from a certain Turing degree:

Definition 1.2.2. A computable structure $\mathcal{C}$ is $\mathbf{d}$-*categorical* for a Turing degree $\mathbf{d}$ if, for every computable $\mathcal{A}$ such that $\mathcal{A} \cong \mathcal{C}$, there is a $\mathbf{d}$-computable isomorphism from $\mathcal{C}$ to $\mathcal{A}$.

Sometimes, it is of natural interest to see if $\mathbf{d}$ is the least such degree (i.e. it exactly describes the difficulty of computing isomorphisms between copies of the structure). The following definition of a "degree of categoricity" for a computable structure was first introduced by Fokina, Kalimullin and Miller in [15] :

Definition 1.2.3. A Turing degree $\mathbf{d}$ is the *degree of categoricity* of a computable structure $\mathcal{C}$ if $\mathbf{d}$ is the least degree such that $\mathcal{C}$ is $\mathbf{d}$-computably categorical. The degree $\mathbf{d}$ is a degree of categoricity if it is the degree of categoricity of some computable structure.

Clearly there are only countably many degrees of categoricity and so there are examples of degrees that are not degrees of categoricity (see [1]). So far, every known example of a degree of categoricity $\mathbf{d}$ has the following additional property: There is a structure $\mathcal{C}$ with computable copies $\mathcal{C}_1$ and $\mathcal{C}_2$ such that not only does $\mathcal{C}$ have degree of categoricity $\mathbf{d}$, but every isomorphism $f : \mathcal{C}_1 \to \mathcal{C}_2$ computes $\mathbf{d}$. If a degree of categoricity has this property, we say it is a *strong degree of categoricity*. Recently, Bazhenov, Kalimullin and Yamaleev [6] have shown that there is a c.e. degree $\mathbf{d}$ and a structure $\mathcal{A}$ with degree of categoricity

$\mathbf{d}$, but $\mathbf{d}$ is not a strong degree of categoricity for $\mathcal{A}$. Csima and Stephenson [14] have shown that there is a structure that has a degree of categoricity but no strong degree of categoricity.

For every computable ordinal α, $\mathbf{0}^{(\alpha)}$ is the degree of categoricity of some tree structure ([13],[27]). However, the trees used were not optimal in the sense that they do not have the least possible rank. In Chapter 2, we realize the same degrees using trees of around half the ranks of those used before. In fact, our trees are of minimal ranks for that purpose.

Fokina, Kalimullin, and Miller [15] showed that, for any $n < \omega$, if $\mathbf{d} \geq \mathbf{0}^{(n)}$ is the Turing degree of a difference of computably enumerable sets (d.c.e.) relative to $\mathbf{0}^{(n)}$, then $\mathbf{d}$ is a degree of categoricity. Also in the same paper they showed that $\mathbf{0}^{(\omega)}$ is a degree of Categoricity. Later, Csima, Franklin, and Shore [13] expanded the result to every computable successor ordinal α. They showed that, for every computable successor ordinal α, every degree d.c.e. in and above $\mathbf{0}^{(\alpha)}$ is a degree of categoricity. Also in the same paper they showed that, for every computable α (not only successors), $\mathbf{0}^{(\alpha)}$ is a degree of categoricity. Csima and Ng have announced a proof that every Δ_2 degree is a degree of categoricity. In Chapter 3, we present our joint work with Csima, Deveau and Harrison-Trainor which fills in a gap above limit ordinals that was missing from the work in [13] making progress towards Question 5.1 of that paper. The question was: Is every degree that is n-c.e. in and above a degree of the form $\mathbf{0}^{(\alpha)}$ for a computable limit ordinal α and $n < \omega$ a (strong) degree of categoricity? We show here that, for every computable limit ordinal α, if $\mathbf{d}$ a degree c.e. in and above $\mathbf{0}^{(\alpha)}$, then $\mathbf{d}$ is a strong degree of categoricity.

1.2.2 The Isomorphism Problem

Whereas computable categoricity measures the complexity of isomorphisms between copies of structures, it is also of interest sometimes to measure the complexity of the question whether two given computable structures over the same signature are isomorphic. It makes sense to study such a question for known classes of computable structures. Given a class of structures K, the isomorphism problem for K is the set $I(K) = \{(n, m) : \mathcal{M}_n, \mathcal{M}_m \in K, \mathcal{M}_n \cong$

$\mathcal{M}_m\}$ where $(\mathcal{M}_n)_{n\in\omega}$ is some computable enumeration of all computable structures (on a fixed computable language). Asking whether two given structures from K are isomorphic is exactly the same as asking whether the ordered pair formed from their indices belongs to $I(K)$.

It is well known that the isomorphism problem for computable structures is complete for the first existential level of the analytical hierarchy Σ^1_1. In fact, the Σ^1_1-completeness is known to hold for many subclasses of computable structures (see [10], [17], [33], [34], [21]). As a consequence, such subclasses do not have a "good classification" (they do not have a hyperarithmetical Friedberg enumeration). For some other classes, as we shall see, the complexity of the isomorphism problem is relatively low (e.g. Π_3); hence, they have a good classification.

In Chapter 4, we completely describe the complexity of the isomorphism problem for the class of computable trees of rank α for every computable ordinal α. We also describe the complexity of the isomorphism problem for a class of pregeometries which generalizes a number of well known examples. In [20], Harrison-Trainor, Melnikov and Montalbán gave a sufficient condition for a structure with some notion of independence to have a computable presentation with a computable basis ("good" copy) and a another computable presentation with no computable basis ("bad" copy). They applied the condition to differentially closed, real closed, and difference closed fields. The condition also implied classical results on vector spaces, algebraically closed fields, torsion-free groups and Archimedean ordered abelian groups. In [8], Calvert studied the isomorphism problem for classes that admit a notion of independence. We observed that the ideas in [20] are applicable to the classes studied by Calvert.

1.2.3 The Degree of a Structure and the Turing Ordinal

One natural way to measure the complexity of a structure is by assigning to it the Turing degree of its atomic diagram (regarded as a set of natural numbers). In this sense, different copies of a structure may have different degrees without any relationship that relates them (even without changing the universe). For

example, for every Turing degree $\mathbf{d}$, one can construct a copy of the usual linear order on the natural numbers but with degree $\mathbf{d}$. Indeed, suppose D is a set of degree $\mathbf{d}$ and consider the linear order with universe $\mathbb{N}$ and $<$ defined as per usual except that $2n+1 < 2n$ if $n \in D$.

It is desirable to assign degrees in a way that is invariant up to isomorphism. We have the following notion:

Definition 1.2.4 (Jockusch)**.** For a structure $\mathcal{A}$:

(1) The *degree spectrum* of $\mathcal{A}$ is

$$\mathbf{Spec}(\mathcal{A}) = \{\mathbf{deg}(D(\mathcal{B})) : \mathcal{B} \cong \mathcal{A},\ \mathcal{B} \text{ has a computable universe }\};$$

and

(2) $\mathcal{A}$ has degree $\mathbf{d}$ if $\mathbf{d}$ is the least member of $\mathbf{Spec}(\mathcal{A})$.

A structure is trivial if there is a finite tuple in the domain of that structure such that any permutation of the domain which fixes that tuple is an isomorphism of the structure. Knight showed that the degree spectrum of a trivial structure is a singleton, whereas that of a non-trivial structure is upward closed [24]. Therefore, if a non-trivial structure has degree $\mathbf{d}$, then its degree spectrum is the cone of Turing degrees above $\mathbf{d}$.

In case that a non-trivial structure does not have a degree, it is perhaps possible that we are still capable of assigning a degree of complexity to that structure using the jumps (or iterated jumps) of the degrees in its spectrum. Consider the following generalization:

Definition 1.2.5 (Jockusch)**.** For any computable ordinal α, $\mathcal{A}$ has α^{th} *jump degree* $\mathbf{d}$ if $\mathbf{d} = \min\{\mathbf{deg}(D(\mathcal{B})^{(\alpha)}) : \mathcal{B} \cong \mathcal{A},\ \mathcal{B} \text{ has a computable universe }\}$. When $\alpha = 0$, we know that $\mathbf{d}$ is the degree of the structure $\mathcal{A}$.

Given a class of structures K, and any computable ordinal α, one can ask which degrees can be realized as the α^{th} jump degree of a structure in K. For example, for the case $\alpha = 0$, if K is the class of graphs, every degree is realizable as the degree of a structure in K (see [35]). If K is the class of linear

orderings, we have that only $\mathbf{0}$ is the degree of a structure in K (see [35]). This suggests that coding information into linear orders is "harder" than it is into graphs.

What about coding into the first jums? In other words, is every degree $\geq \mathbf{0}'$ the first jump degree of some linear order? It turns out that only $\mathbf{0}'$ can be realized as the first jump degree of a linear order (see [24]). But, any degree $\geq \mathbf{0}^{(2)}$ is the second jump degree of a linear order [3]. Finding the ordinal $\alpha = 2$ at which we started being capable of coding any degree gave us, in some sense, a measurement of how much harder linear orderings are than graphs when it comes to coding information.

The idea of coding information into jumps motivated Jockusch and Soare to introduce the following definition.

Definition 1.2.6 (The Turing Ordinal)**.** Let T be a first-order theory with continuum many pairwise non-isomorphic countable models. We call a computable ordinal γ the *Turing ordinal* (abbreviated as *TO*) of T if:

(1) every degree $\geq \mathbf{0}^{(\gamma)}$ is the γ^{th} jump degree of a model of T, and

(2) for all $\eta < \gamma$, the only possible η^{th} jump degree of a model of T is $\mathbf{0}^{(\eta)}$.

Note that the definition is for a theory T. Instead, we will be talking about the Turing Ordinal for a class of structures (which could be chosen to be the class of models of some theory T).

One natural question that arises from the definition is whether the Turing Ordinal always exists. As the reader may have noticed, there are two conditions that must hold true. The first condition says that every degree above the γ^{th} jump is the γ^{th} jump degree of some model, whereas the second condition says that, for $\eta < \gamma$, **only** one degree (which is $\mathbf{0}^{(\eta)}$) can be the η^{th} jump degree of some model. There are two events here; one of them starts at γ while the other stops at γ. There is no guarantee that there is such an ordinal γ. In fact, Montalbán gave an example of a theory for which the Truing ordinal does not exist (see [32]).

To understand the Turing ordinal better, we break down its definition into two auxiliary definitions.

The Upper Turing Ordinal: Let K be a class of first-order structures. We call a computable ordinal β the *upper Turing ordinal (abbreviated as UTO)* of K if it is the least ordinal such that the first condition of the *TO* definition holds. More precisely:

(1) every degree $\geq \mathbf{0}^{(\beta)}$ is the β^{th} jump degree of a structure in K, and

(2) for all $\eta < \beta$, there exists a degree $\mathbf{d} \geq \mathbf{0}^{(\eta)}$ which is not the η^{th} jump degree of a structure in K.

The Lower Turing Ordinal: Let K be a class of first-order structures. We call a computable ordinal α the *lower Turing ordinal (abbreviated as LTO)* of K if it is the greatest ordinal such that the second condition of the *TO* definition holds. More precisely:

(1) there exists a degree $\mathbf{d} \neq \mathbf{0}^{(\alpha)}$ such that $\mathbf{d}$ is the α^{th} jump degree of a structure from K, and

(2) for all $\eta < \alpha$, the only possible η^{th} jump degree of a structure from K is $\mathbf{0}^{(\eta)}$.

It is clear that the Turing ordinal exists if and only if the upper and lower Turing ordinals exist and are equal. In Chapter 5, for any computable ordinals α, β such that $\alpha < \beta$, we give examples of "nice" classes of structures such that *UTO* = β and *LTO* = α.

Chapter 2

Degrees of Categoricity of Trees

The work in this chapter, as well its consequences in Chapter 4, have been accepted for publication by the Mathematical Logic Quarterly (MLQ)[28].

In this chapter, we study a specific class of structures which is the class of computable trees. By a tree T we mean a universe (set of nodes or vertices) together with a strict partial ordering $\prec$ such that:

(1) For all x, the set of predecessors of x is finite and well ordered by $\prec$.

(2) T contains a unique least element under $\prec$ called the *root*.

We will say for $x, y \in T$ that y is an *immediate successor of* x (or that x is the *immediate predecessor* of y) if $x \prec y$, and for no z in T, $x \prec z \prec y$. Note that for every y except the root, there exists a unique immediate predecessor because of condition 1.

For a node x in a tree T, we define its level as $Level_T(x) = |\{y \in T : y \prec x\}|$. All trees we use throughout this thesis have no infinite paths, and therefore the level of a node is always a natural number. The height of a tree T, $h(T) = sup_{x \in T}(Level_T(x) + 1)$. This means that a tree of height 1 is a single point (trivial case). The least possible height is 1 because a tree as a structure is assumed to be non-empty. For our purposes here, it will be rather more convenient to use the concept of a *rank* instead of height.

For a tree of height $\leq \omega$ that has no infinite path, each terminal node is assigned rank 0, and each nonterminal node x has as its rank ($rk(x)$) the supremum of the ordinals $rk(y) + 1$ where y ranges over all the immediate successors (equivalently, over all successors) of x. The rank of a tree is the rank of its root node. For finite height trees, the rank plus one equals the height. All countable ordinals can be ranks of countable trees, and all computable ordinals can be ranks of computable trees.

For $u \in T$, we will use the notation $T[u]$ to refer to the subtree of T which has the node u as a root.

By a computable tree we mean that both the universe T and the relation $\prec$ are computable (recall Definition 1.2.1 of a computable structure). Note that according to this definition, in a computable tree, deciding whether y is an immediate successor of x is not computable (it is co-c.e.). Although this is the general case, all of the trees we are going to use in our proofs have a computable immediate successor relation, and so our results are also valid in case we look at $\prec$ as the edge relation (on the tree as a graph). In general different ways of defining $\prec$ give computationally different properties.

It is not hard to see that, for any computable ordinal α, a computable tree of rank $\alpha + 1$ is $\mathbf{0}^{(2\alpha)}$-categorical if α is finite, and $\mathbf{0}^{(2\alpha+1)}$-categorical if α is infinite (Proposition 2.1.2 below). From this it is natural to wonder, if the oracle $\mathbf{0}^{(2\alpha)}$ (or $\mathbf{0}^{(2\alpha+1)}$ in the infinite case) is in general necessary or more than enough. We answer here that it is necessary (Theorem 2.2.11).

2.1 Preparation

Given two isomorphic trees S, T (all our trees are assumed computable unless otherwise stated), if we want to define an isomorphism between them, then we need to know which level 1 nodes in S should be mapped to level 1 nodes in T. If a level 1 node u from S is mapped to v from T, then this means that the subtrees $S[u]$ and $T[v]$ are isomorphic. We continue defining the isomorphism by matching level 1 nodes in $S[u]$ and $T[v]$ and so on. To know how complex the isomorphism is, we need to know how complex it is to determine the level 1 nodes in S, T, and we also need to

know how complex it is to decide for subtrees $S[u]$, $T[v]$ (rooted at level 1) if they are isomorphic (clearly there is induction going on there). It is straightforward to see that, in general, for $x \in T$ and $n \in \mathbb{N}$, the question "$Level_T(x) = n$?" is computable in $\mathbf{0}'$. More clearly, $Level_T(x) = n$ if and only if $(\exists x_1 ... x_n)[x_1 \prec \ldots \prec x_n \prec x] \wedge (\forall x_1 ... x_{n+1})[\neg(x_1 \prec \ldots \prec x_{n+1} \prec x)]$.

Let us now give a detailed proof of the fact that, for a computable ordinal α, a computable tree of rank $\alpha + 1$ is $\mathbf{0}^{(2\alpha)}$-categorical if α is finite, $\mathbf{0}^{(2\alpha+1)}$-categorical if α is infinite. First we need the following lemma:

Lemma 2.1.1. *Checking whether two computable trees of rank α are isomorphic is $\Pi_{2\alpha}$-uniformly.*

Proof. We use induction on α. The base case $\alpha = 0$ is easy, each tree is only a single point. Assume the lemma holds for all $\beta < \alpha$. For any two computable trees S, T of rank α, they are both isomorphic if and only if for every level 1 node u in S, the isomorphism type of $S[u]$ exists among the subtrees of S with roots at level 1 exactly as many times as in T. In symbols we can write this as:

$$\forall u \Bigg[u \in S \wedge Level_S(u) = 1 \Rightarrow$$

$$\forall k \Bigg(\exists \bar{u} \Big(|\bar{u}| = k \wedge \bigwedge_{i \neq j < k} u_i \neq u_j \wedge \forall i < k (Level_S(u_i) = 1 \wedge S[u_i] \cong S[u]) \Big) \Rightarrow$$

$$\exists \bar{v} \Big(|\bar{v}| = k \wedge \bigwedge_{i \neq j < k} v_i \neq v_j \wedge \forall i < k (Level_T(v_i) = 1 \wedge T[v_i] \cong S[u]) \Big) \Bigg) \Bigg] \wedge$$

$$\forall v \Bigg[v \in T \wedge Level_T(v) = 1 \Rightarrow$$

$$\forall k \Bigg(\exists \bar{v} \Big(|\bar{v}| = k \wedge \bigwedge_{i \neq j < k} v_i \neq v_j \wedge \forall i < k (Level_T(v_i) = 1 \wedge T[v_i] \cong T[v]) \Big) \Rightarrow$$

$$\exists \bar{u} \Big(|\bar{u}| = k \wedge \bigwedge_{i \neq j < k} u_i \neq u_j \wedge \forall i < k (Level_S(u_i) = 1 \wedge S[u_i] \cong T[v]) \Big) \Bigg) \Bigg].$$

This is a conjunction of two formulas each is of the form:

$$\forall u[\Pi_1 \Rightarrow \forall k(\exists \Pi_{2\beta_{u,k}} \Rightarrow \exists \Pi_{2\beta_{u,k}})],$$

where $\beta_{u,k} < \alpha$ for all $u, k \in \omega$.

Such a formula is equivalent to one of the form:

$$\forall u[\Sigma_1 \vee \forall k(\forall \Sigma_{2\beta_{u,k}} \vee \exists \Pi_{2\beta_{u,k}})].$$

This can be written as $\forall u \forall k[\Sigma_1 \vee \forall \Sigma_{2\beta_{u,k}} \vee \exists \Pi_{2\beta_{u,k}}]$ which is the same as $\forall u \forall k[\Pi_{2\beta_{u,k}+1} \vee \exists \Pi_{2\beta_{u,k}}]$. We can pull the existential quentifier outside the brackets, and regard the universal quantifier at the beginning as an infinitary conjunction (obviously of a c.e. set of formulas) to obtain

$$\bigwedge_{u,k \in \omega} \exists \Pi_{2\beta_{u,k}+1}.$$

Since, for every $u, k \in \omega$, $\beta_{u,k} < \alpha$, we must have that $2\beta_{u,k} + 1 < 2\alpha$. Hence, our formula is $\Pi_{2\alpha}$.

To explain some parts more, note that, for a node u, deciding whether it is of level 1 in a computable tree is equivalent to answering a Π_1 question: "Is it the case that, for all v different from u, and from the root r, there is no v such that $r \prec v \prec u$?". Moreover, by the induction hypothesis, we can decide whether $S[u_i] \cong T[v]$ (similarly whether $T[v_i] \cong S[u]$) by answering a $\Pi_{2\beta}$ question for some $\beta < \alpha$. This can be done uniformly in u_i and v; also uniformly (within Π_α) we can find the ranks of the subtrees $S[u_i]$ and $T[v]$. The latter can be checked by a simple inductive argument. □

Note that in the case $\alpha = 1$, what we did is exactly comparing the sizes of the trees which can be done using $\mathbf{0}^{(2)}$.

Proposition 2.1.2. *For any two computable copies S, T of a tree of computable rank $\alpha + 1$, there exists an isomorphism from S to T which is $\mathbf{0}^{(2\alpha)}$-computable if α is finite, and $\mathbf{0}^{(2\alpha+1)}$-computable if α is infinite. In other words, a computable tree of rank $\alpha + 1$ is $\mathbf{0}^{(2\alpha)}$-categorical if α is finite, and $\mathbf{0}^{(2\alpha+1)}$-categorical if α is infinite. Moreover, if α is a limit ordinal or 0, a computable tree of rank α is $\mathbf{0}^{(2\alpha)}$-categorical.*

Proof. Let S and T be two copies of a tree of rank $\alpha + 1$. We define an isomorphism $f : S \to T$ going back and forth. We go through level 1 in S until we find the first u not in the domain of f yet, and we look for the first v in level 1 in T which is not in range f yet such that $S[u] \cong T[v]$. Set $f(u) = v$. Conversely, for the first y in level 1 in T not in the range of f yet, look for the first x in level 1 in S not in the domain of f yet such that $S[x] \cong T[y]$. Set $f(x) = y$. By the lemma, for finite (infinite) α, $\mathbf{0}^{(2\alpha)}$ ($\mathbf{0}^{(2\alpha+1)}$) can decide uniformly for us whether $S[u] \cong T[v]$ for all u, v. Therefore, for finite (infinite) α, $f \leq_T \mathbf{0}^{(2\alpha)}$ ($\mathbf{0}^{(2\alpha+1)}$).

For the case when S and T are copies of a tree of rank α (limit or 0), if α is a limit ordinal, then every subtree with root in level 1 in S or T has rank $< \alpha$. So, again, $\mathbf{0}^{(\alpha)}$ ($= \mathbf{0}^{(2\alpha)}$) can decide uniformly for us whether $S[u] \cong T[v]$ for all u, v. The case $\alpha = 0$ is trivial.

Note that it looks like we have not considered how the complexity of determining the root and level 1 nodes is involved in the complexity of f just mentioned. Everything is fine because determining each can be done using $\mathbf{0}'$ (x is the root if and only if $\neg(\exists y)(y \prec x)$, and x is a level 1 node if and only if $(\exists y)(y \prec x) \wedge \neg(\exists y)(\exists z)(y \prec z \prec x)$). So for $\alpha > 1$ it does not affect the complexity $\mathbf{0}^{(2\alpha)}$ (or $\mathbf{0}^{(2\alpha+1)}$). For $\alpha = 1$, we just map the root to the root and the rest of the vertices can be mapped computably. So f can be computable. □

Now after seeing the last proposition, it is natural to ask if we necessarily need the 2α-th (or $(2\alpha + 1)$-th) jump, not less. More precisely, we are trying to establish for every finite (infinite) α the existence of two copies of a tree of rank $\alpha + 1$ such that any isomorphism between the two copies can compute $\mathbf{0}^{(2\alpha)}$ ($\mathbf{0}^{(2\alpha+1)}$). Also if α is a limit, there exists a tree of rank α with two copies such that any isomorphism between the two copies can compute $\mathbf{0}^{(2\alpha)}$. Indeed, we could prove this for all computable ordinals α (Theorem 2.2.3). The case $\alpha = 0$ is trivial and the case $\alpha = 1$ is easy. Let us have a look now at $\alpha = 2$ to see how things go.

Proposition 2.1.3. *There exists a tree of rank 2 with computable copies S_2, T_2 such that every isomorphism between the two copies can compute $\mathbf{0}^{(2)}$.*

Proof. The existence of such a tree can be proved by just directly defining the

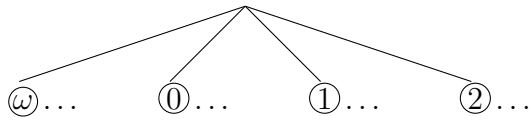


Figure 2.1: The isomorphism type of S_2 (or T_2).

copies. Here, as well as in all our next results, we will describe our trees as sets of finite sequences (which can be effectively coded by natural numbers), and $\prec$ will be the initial segment relation.
Let S_2 be the tree with root $()$ (the empty sequence) and level 1 nodes (i) for all $i \in \omega$. For level 2, include (i, s) if and only if $W_{i,s+1} \neq W_{i,s}$ (See Figure 2.1). Note that W_i is infinite if and only if (i) has infinitely many children (branches). Also note that in S_2, from the padding lemma and the fact that there are c.e. sets of every size, we have infinitely many subtrees of every size $\leq \omega$ with roots in level 1 in S_2.

Now let us define the tree T_2. As usual, T_2 has root $()$. Let level 1 consist of all nodes (i), $i \in \omega$. For level 2, include $(2i, j)$ for all $i, j \in \omega$, and $(2\langle i, k\rangle + 1, j)$ for $i, k \in \omega$ and $j < i$. Note that $(2\langle 0, k\rangle + 1)$ is a dead end for all k. Clearly both trees are isomorphic. Also it is clear that every isomorphism f computes $\mathbf{Inf} = \{e : W_e \text{ is infinite }\}$ because $r \in \mathbf{Inf}$ if and only if $f(r)$ is even. $\mathbf{Inf}$ is Π_2-complete and so f can compute $\mathbf{0}^{(2)}$.

The isomorphism type of S_2 (or T_2) looks like:

Where $\textcircled{N}$ is the tree type of rank 1 with N children for $N > 0$, and $\textcircled{0}$ is the tree of rank 0 (a single dot).

□

The construction above was simple. The case $\alpha \geq 2$ is a lot harder; we will use a dynamic way to construct our trees (movable marker technique, see [36]). We will do the case when α is finite first. Before we go for such construction in the next section, we recall some representation theorems.

Consider the following notation (where U is some oracle):

$$\mathbf{Fin}^U = \{x : W_x^U \text{is finite}\} \qquad \mathbf{Inf}^U = \{x : W_x^U \text{is infinite}\}$$
$$\mathbf{Cof}^U = \{x : W_x^U \text{is cofinite}\} \qquad \mathbf{Tot}^U = \{x : W_x^U = \omega\}$$

We have the following:

Lemma 2.1.4. *(1)* $\Sigma_2^U \leq_1 \mathbf{Fin}^U$ *(equivalently,* $\Pi_2^U \leq_1 \mathbf{Inf}^U$*).*

(2) If $A \in \Sigma_3^U$*, then there is a computable function (not merely* U*-computable)* h *such that, for all* $x \in \omega$*:*

$$x \in A \Leftrightarrow (\exists y)[h(x,y) \in \mathbf{Tot}^U \wedge (\forall z \neq y) h(x,z) \in \mathbf{Fin}^U],$$

$$x \notin A \Leftrightarrow (\forall y)[h(x,y) \in \mathbf{Fin}^U].$$

Note that when $x \in A$*, there is a unique* y *such that* $W_{h(x,y)}^U$ *is infinite.*

Proof. This is a straightforward relativization of results in [36](pages 86-91). The function h is computable, not just U-computable, because its definition is independent of the oracle U (we use the relativized s-m-n Theorem). □

2.2 Results

Our aim in this section is to generalize Proposition 2.1.3. We mentioned before that this will not be as easy and we will need some dynamic procedure. For the reader to see the issue, let us look at the most obvious way to generalize the result to rank 3 and it will be clear then why such obvious generalization does not seem to work. Looking at the proof of Proposition 2.1.3, we have built two trees S, T such that T was "nice" while S was "hard". The tree T was nice in the sense that it was not built to "code" a certain set; it was just an evident way to describe a member of the isomorphism type in detail. On the other hand S was "hard" in the sense that it was built through the knowledge of some

specific set with certain complexity to code that set. A level 1 node in S was the father of infinitely many children if and only if the node corresponded to a member of $\mathbf{Inf}$. This means that S was built to code the set $\mathbf{Inf}$ in some sense. We will try to do the same for rank 3, and we will keep the nice copy denoted T and the hard copy denoted S. We will assume D is some Σ_4 set (which we can take later to be $\varnothing^{(4)}$) and we will build S so it codes D in information about level 1 nodes.

We know that there exists some Σ_2 relation R such that $x \in D$ if and only if $\exists y \forall z R(x, y, z)$. By Lemma 2.1.4(1), R is one-one reducible to $\mathbf{Fin}$. This gives the intuition that we want to end up having that a level 1 node x is the root of some sort of "finite type" subtree of rank 2 if and only if $x \in D$, or the root of some "infinite type" subtree if and only if $x \notin D$. We later will give precise definitions to what we mean by finite and infinite types; now we are just helping the reader understand the picture.

The direct way to represent D at this point will be as follows: For some computable function $g(x, y, z)$,

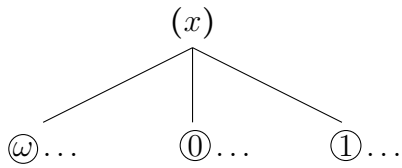
$$x \in D \Leftrightarrow \exists y \forall z W_{g(x,y,z)} \text{ is finite;}$$

$$x \notin D \Leftrightarrow \forall y \exists z W_{g(x,y,z)} \text{ is infinite.}$$

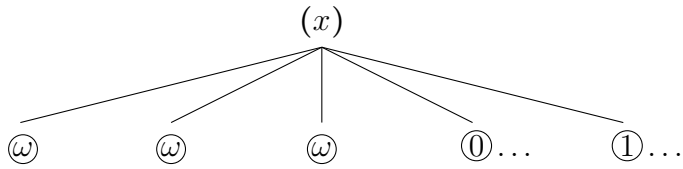
Now if we want to build the tree S of rank 3 from this information, it seems that we should try something like: Root and level 1 as usual. Level 2 includes $(x, \langle y, z \rangle)$ for all y, z, and level 3 will include $(x, \langle y, z \rangle, s)$ whenever $W_{g(x,y,z),s+1} \neq W_{g(x,y,z),s}$. Unfortunately we end up getting a "bad" tree in the sense that there is no way to distinguish whether x belongs to D or not based on the shape of the subtree below it.

Imagine now we could find a way of representing Σ_4 sets that makes the tree S look "good"; say for example we could make the subtrees of the type $\textcircled{n}$ for each finite n always appear infinitely often and $\textcircled{\omega}$ appears either infinitely many or finitely many (and when it happens finitely many, it will not matter how many exactly).

When $x \notin D$:



When $x \in D$:



If we accomplish this, we basically would have solved the problem. We will make the nice copy T so it has the level 1 even nodes, say, as roots to the first figure above and the odd nodes are roots to the second figure and so, an arbitrary isomorphism from S to T will take x to an even number if and only if $x \notin D$ and D will be computable from any isomorphism.

In what follows we establish such a helpful representation result for Σ^U_4 sets where U can be any oracle.

Lemma 2.2.1. *Let D be a Σ^U_4 set. Then, there exists a computable function $\tilde{h}$ such that for all x:*

(1) $x \in D \Leftrightarrow (\exists y)(\forall z)[\tilde{h}(x,y,z) \in \mathbf{Fin}^U]$;

(2) $(\forall y)[(\forall z)\tilde{h}(x,y,z) \in \mathbf{Fin}^U \Rightarrow (\forall y' \geq y)(\forall z)\tilde{h}(x,y',z) \in \mathbf{Fin}^U]$; *and*

(3) $(\forall y)[(\exists z)\tilde{h}(x,y,z) \in \mathbf{Inf}^U \Rightarrow (\exists! z)\tilde{h}(x,y,z) \in \mathbf{Inf}^U]$.

Proof. Since D is a Σ^U_4 set, then there exists some Σ^U_3 relation R such that $x \in D$ if and only if $\exists y \neg R(x,y)$. By Lemma 2.1.4(2), there is a computable function h such that

$$R(x,y) \Leftrightarrow (\exists! z)(W^U_{h(x,y,z)} = \omega \wedge (\forall w \neq z)[W^U_{h(x,y,w)} \text{ is finite}]) \quad (*),$$

$$\neg R(x,y) \Leftrightarrow (\forall z)(W^U_{h(x,y,z)} \text{ is finite}).$$

That is, we have

$$x \in D \Leftrightarrow (\exists y)(\forall z)(W^U_{h(x,y,z)} \text{ is finite}),$$

$$x \notin D \Leftrightarrow (\forall y)(\exists! z)(W^U_{h(x,y,z)} = \omega \wedge (\forall w \neq z)[W^U_{h(x,y,w)} \text{ is finite}]).$$

It is not hard to see that, for any x (not necessarily outside D) if y is such that for some z, $W^U_{h(x,y,z)}$ is infinite, then such z is unique. Indeed, if such z exists, then it is not the case that $\neg R(x,y)$, i.e., $R(x,y)$ and so z is unique by $(*)$. This means that condition 3 alone is satisfied if $\tilde{h}$ is taken to equal h (clearly condition 1 also valid). We get $\tilde{h}$ to satisfy condition 2 by modifying h in a way that will keep properties 1,3 valid.

To describe $\tilde{h}$, we use the movable marker method. Our marker position at each stage will be determined based on a U-computable function $r^U : \mathbb{N}^3 \to \mathbb{N} \cup \{+\infty\}$. For fixed x, y, at every stage s, $r^U(x,y,s)$ tries to land on the least value z such that $W^U_{\tilde{h}(x,y-1,z)}$ is infinite. We define U-c.e. sets uniformly in x, y, z (construction is also uniform in U, i.e., it is the same using any set in place of U).

For all x, y set $r^U(x,y,0) = +\infty$.

First, for $y = 0$, set $W^U_{\tilde{h}(x,0,z)} = W^U_{h(x,0,z)}$, and set $r^U(x,0,s+1) = \mu l\{l < s+1 : W^U_{h(x,0,l),s+1} \neq W^U_{h(x,0,l),s}\}$ if such l exists, and otherwise $r^U(x,0,s+1) = +\infty$.

Fix $y > 0$ and assume $W^U_{\tilde{h}(x,y-1,z)}$ is defined for all x, z. We will define $W^U_{\tilde{h}(x,y,z)}$ in stages.

Stage 0: $W^U_{\tilde{h}(x,y,z),0}$ is empty.

Stage $s+1$: Assume $W^U_{\tilde{h}(x,y,z),s}$ as well as $r^U(x,y,s)$ are now defined for all x, z. If $\pi_2(z) = r^U(x,y,s)$ (otherwise do nothing), set $W^U_{\tilde{h}(x,y,z),s+1} = W^U_{\tilde{h}(x,y,z),s} \cup W^U_{h(x,y,\pi_1(z)),s}$ (notice the uniformity of the definition of $r^U(x,y,s)$), and set $r^U(x,y,s+1)$ to be the least $l < s+1$ (if it exists, otherwise $r^U(x,y,s+1) = +\infty$) such that $W^U_{\tilde{h}(x,y-1,l),s+1} \neq W^U_{\tilde{h}(x,y-1,l),s}$.

End of description.

Now we show that $\tilde{h}$ does what we want.

Case 1: $x \in D$. Choose y to be least such that $(\forall z)(W^U_{h(x,y,z)}$ is finite$)$. If $y = 0$, then $W^U_{\tilde{h}(x,y,z)} = W^U_{h(x,y,z)}$ which is finite for all z. If $y > 0$, then for every z, $W^U_{\tilde{h}(x,y,z)} \subseteq W^U_{h(x,y,\pi_1(z))}$ which is also finite. As a consequence, for every natural value l, it happens that $r^U(x,y+1,s) = l$ for only finitely many s. This will make $W^U_{\tilde{h}(x,y+1,z)}$ finite for all z because $W^U_{\tilde{h}(x,y+1,z)}$ gains elements at a stage $s+1$ only if $\pi_2(z) = r^U(x,y+1,s)$. Inductively, we have for all $y' \geq y$ that $W^U_{\tilde{h}(x,y',z)}$ is finite for all z.

To complete verifying this case, we need to show that if $y > 0$ is like above, then, for every $y' < y$, there is a unique z' such that $W^U_{\tilde{h}(x,y',z')}$ is infinite. For $y' = 0$ this is direct because $W^U_{\tilde{h}(x,0,z)} = W^U_{h(x,0,z)}$. Assume $0 \leq y' < y$. Note that, for every z, $W^U_{\tilde{h}(x,y',z)} \subseteq W^U_{h(x,y',\pi_1(z))}$, and so the former is possibly infinite whenever the latter is. There is a unique z'' such that $W^U_{h(x,y',z'')}$ is infinite, and so for only the z-values such that $\pi_1(z) = z''$, $W^U_{\tilde{h}(x,y',z)}$ is possibly infinite. But we need more to verify, we need a unique z such that $W^U_{\tilde{h}(x,y',z)}$ is possibly infinite (which is actually infinite). Indeed, $z' = \langle z'', l\rangle$ is that unique z where $l = \liminf_s r^U(x, y', s)$ (the unique l such that $l = r^U(x, y', s)$ for infinitely many s). The last sentence can be verified by induction on $y' < y$: $y' = 0$ is clear. Assume there is a unique l_1 such that $l_1 = r^U(x, y' - 1, s)$ for infinitely many s, and $z_1 = \langle z''_1, l_1\rangle$ (where z''_1 is the unique such that $W^U_{h(x,y'-1,z''_1)}$ is infinite) is the unique such that $W^U_{\tilde{h}(x,y'-1,z_1)}$ is infinite. By the definition of r^U, z_1 is the unique value such that $z_1 = r^U(x, y', s)$ for infinitely many s. At every stage s, all (and only) the elements of $W^U_{h(x,y',\pi_1(z)),s}$ get enumerated into $W^U_{\tilde{h}(x,y',z),s+1}$ only when $\pi_2(z) = z_1$. Whence, $z' = \langle z'', z_1\rangle$ (where z'' is the unique such that $W^U_{h(x,y',z'')}$ is infinite), is the unique such that $W^U_{\tilde{h}(x,y',z')}$ is infinite.

Case 2: $x \notin D$. We show by induction on y that for every y, $\liminf_s r^U(x, y, s)$ exists and that there is unique $\tilde{z}_y$ such that $W^U_{\tilde{h}(x,y,\tilde{z}_y)}$ is infinite. Indeed, first we know that for every y there is unique z (call it z_y) such that $W^U_{h(x,y,z_y)}$ is infinite. For $y = 0$, clearly $\liminf_s r^U(x, y, s) = z_0$ (we have $\tilde{z}_0 = z_0$) and $W^U_{\tilde{h}(x,0,\tilde{z}_0)} = W^U_{\tilde{h}(x,0,z_0)}$ which is infinite. Assume now $\liminf_s r^U(x, y, s) = l$ exists and that $W^U_{\tilde{h}(x,y,\tilde{z}_y)}$ is infinite (assume $\tilde{z}_y$ is unique). Then we have that $\liminf_s r^U(x, y+1, s) = \tilde{z}_y$. Take $\tilde{z}_{y+1} = \langle z_{y+1}, l\rangle$, then clearly $W^U_{\tilde{h}(x,y+1,\tilde{z}_{y+1})} = W^U_{h(x,y+1,z_{y+1})}$ which is infinite. Notice that $W^U_{\tilde{h}(x,y+1,z)}$ is infinite only if $W^U_{h(x,y+1,\pi_1(z))}$ is infinite and $\pi_2(z) = r^U(x, y + 1, s)$ for infinitely many s. The latter happens only when $\pi_2(z) = \tilde{z}_y$. Hence, $z = \tilde{z}_{y+1}$ is the only value making $W^U_{\tilde{h}(x,y+1,z)}$ infinite. We have that for all y, $\liminf_s r(x, y + 1, s) = \tilde{z}_y$.

□

Now we can build our general case trees in a good shape in the sense we discussed before Lemma 2.2.1. Let us first introduce some new tree notation. For finite $n \geq 1$ and $N \in \mathbb{N} \cup \{\omega\}$, we inductively define type $[n : N]$ trees as

follows.
$[1:N]$ is the tree $\textcircled{N}$ defined before.
$[n:N]$ is the tree such that exactly N of its level 1 vertices are roots of subtrees of type $[n-1:\omega]$ and infinitely many from every type $[n-1:m]$ for every natural m.
For convenience, whenever there is no confusion about the rank, we are going to call the type $[n:\omega]$ "infinite type" while, for finite N, $[n:N]$ is of "finite type".

Lemma 2.2.2. *Given a Σ_{2n+2} set D for some natural $n \geq 0$, there is a uniformly computable sequence of computable trees $(T^x)_{x\in\omega}$ of rank $n+1$ such that for every x*
$x \notin D$ if and only if T^x is of type $[n+1:\omega]$ and
$x \in D$ if and only if T^x is of type $[n+1:m]$ for some finite m.

Proof. We use induction on n. If $n = 0$, we have $D \leq_m \mathbf{Fin}$ say via f. Let T^x be the tree with root (x) and just include (x,s) in level 1 if and only if $W_{f(x),s+1} \neq W_{f(x),s}$. Assume the lemma holds for $n = k$, and let D be a Σ^0_{2k+4} set. By Lemma 2.2.1, there is a Σ^0_{2k+2} relation R such that

(1) $x \in D \Leftrightarrow (\exists y)(\forall z)[R(x,y,z)]$,

(2) $(\forall y)[(\forall z)R(x,y,z) \Rightarrow (\forall y' \geq y)(\forall z)R(x,y',z)]$,

(3) $(\forall y)(\exists z)\neg R(x,y,z) \Rightarrow (\exists! z)\neg R(x,y,z)$.

Now, since for any fixed x, the set $H_x = \{\langle y,z\rangle : R(x,y,z)\}$ is Σ_{2k+2}, so by the induction hypothesis, there exists a uniformly computable sequence of computable trees $T_x^{\langle y,z\rangle}$ such that
$\langle y,z\rangle \notin H_x$ if and only if $T_x^{\langle y,z\rangle}$ is of type $[k+1:\omega]$ and
$\langle y,z\rangle \in H_x$ if and only if $T_x^{\langle y,z\rangle}$ is of type $[k+1:m]$ for some finite m.

Now let T^x be the tree of rank $k+2$ such that every even level 1 vertex $(2\langle y,z\rangle)$ is the root of the tree $T_x^{\langle y,z\rangle}$ and let the odd vertices be roots of infinitely many copies of all the finite types of rank $k+1$. That separation into even and odd vertices is just to guarantee that T^x is of type $[k+2:\omega]$ or $[k+2:m]$.

Now it remains to confirm that our sequence of trees satisfies the conditions.

Case 1: $x \in D$. Let $y = y_0$ be least such that $(\forall z)R(x, y_0, z)$. Then by 2 above, for all $y \geq y_0$, $(\forall z)R(x, y, z)$ and by 3, for all $y < y_0$, $(\exists! z)\neg R(x, y, z)$. This means that for all $y \geq y_0$, the trees $T_x^{\langle y,z\rangle}$ are of the finite type. At the same time, for every $y < y_0$ there exists a unique z_y such that the tree $T_x^{\langle y,z_y\rangle}$ is of type $[n+1 : \omega]$ (for $z \neq z_y$, $T_x^{\langle y,z\rangle}$ is of the finite type). Therefore, the tree T^x is of type $[n+2 : y_0]$.

Case 2: $x \notin D$. In a similar way, one can see that T^x is a tree of type $[n+2 : \omega]$. This is exactly what we want.

□

Now we are ready to prove our main theorem for finite ranks.

Theorem 2.2.3. *For all $n \geq 0$, there exists a tree of rank $n+1$ with strong degree of categoricity $\mathbf{0}^{(2n)}$.*

Proof. From Proposition 2.1.2, all we need to do is to show that there is a tree of rank $n+1$ with computable copies T_{n+1}, S_{n+1} such that every isomorphism between the two copies computes $\mathbf{0}^{(2n)}$. The case $n = 0$ is trivial. Let T^x be a computable sequence of trees like in the previous lemma for $D = \varnothing^{(2n+2)}$. Let S_{n+1} be the tree of type $[n+2 : \omega]$ with root $()$ and every even level 1 vertex $(2x)$ is the root of T^x and let the odd vertices cover infinitely often the possibilities $[n+1 : m]$ for all finite m.

Defining T_{n+1}: simply, make every even level 1 node the root of a tree of type $[n+1 : \omega]$ while keep the odd nodes for infinitely many copies of $[n+1 : m]$, $m \in \mathbb{N}$.

Now, if $f : S_{n+1} \to T_{n+1}$ is an isomorphism, then for all x, $x \in D$ if and only if $f(2x)$ is odd. □

Generalizing the last theorem to rank ω is immediate because of the uniformity of our tree constructions in n:

Corollary 2.2.4. *There exists a tree of rank ω with strong degree of categoricity $\mathbf{0}^{(\omega)}(= \mathbf{0}^{(2\omega)})$.*

Proof. Let T_ω be the tree such that every level 1 node n is the root of the tree T_n obtained in the previous theorem. Similarly let S_ω be the tree such that every level 1 node n is the root of the tree S_n obtained in the previous theorem.

Clearly any isomorphism between T_ω and S_ω uniformly computes $\varnothing^{(2n)}$ for every n, hence, computes $\varnothing^{(\omega)}$. On the other hand, there is an isomorphism between the two copies which is computable from $\mathbf{0}^{(\omega)}$ by Proposition 2.1.2.

□

Note that, in the last corollary, we could also have used the trees from [13] which are Hirschfeldt and White's back-and-forth trees [22](see Definition 2.2.5 below). Also, in [13] (Theorem 3.1), it was shown that there is a computable structure with strong degree of categoricity $\mathbf{0}^{(\omega+1)}$. The structure defined there is a collection of disjoint trees each of rank ω. We can glue those trees to obtain a tree of rank $\omega + 1$ with strong degree of categoricity $\mathbf{0}^{(\omega+1)}$.

Starting at this point, throughout the rest of this chapter, we are going to use the trees $\mathcal{A}_\gamma$, $\mathcal{E}_\gamma$, $\mathcal{L}_k^\gamma$ and $\mathcal{L}_\infty^\gamma$ exactly as defined in Definition 3.1 [22]. We recall the definitions here.

Hirschfeldt and White defined, for each successor ordinal β, a pair of trees $\mathcal{A}_\beta$ and $\mathcal{E}_\beta$ which can be differentiated exactly by β jumps. These trees are called *back-and-forth trees*.

Definition 2.2.5 ([22, Definition 3.1])**.** Back-and-forth trees are defined recursively in β. We take $\mathcal{A}_1$ to be the tree with just a root node and no children, and we take $\mathcal{E}_1$ to be the tree where the root node has infinitely many children, none of which have children. See Fig. 2.2.

Suppose β is a successor ordinal. Define $\mathcal{A}_{\beta+1}$ as a root node with infinitely many children, each the root of a copy of $\mathcal{E}_\beta$, and define $\mathcal{E}_{\beta+1}$ as a root node with infinitely many children, each the root of a copy of $\mathcal{A}_\beta$, and also infinitely many other children, each the root of a copy of $\mathcal{E}_\beta$. See Fig. 2.3.

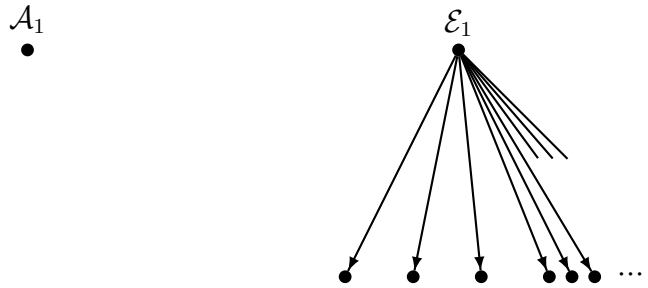


Figure 2.2: $\mathcal{A}_1$ and $\mathcal{E}_1$

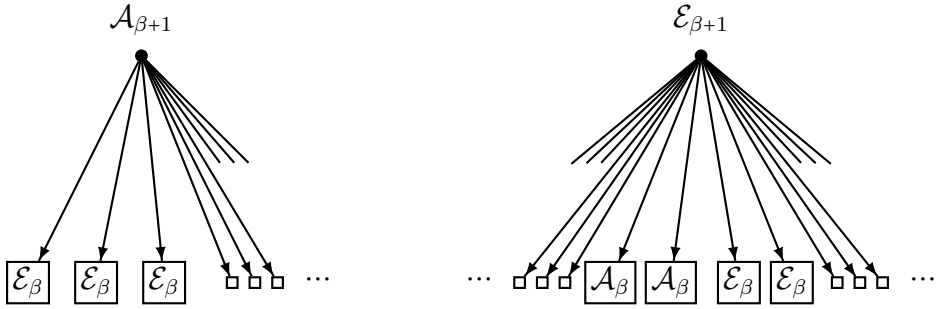


Figure 2.3: $\mathcal{A}_{\beta+1}$ and $\mathcal{E}_{\beta+1}$ when β is a successor ordinal.

Now suppose β is a non-zero limit ordinal, and let $\beta_0, \beta_1, \ldots$ be a fundamental sequence of successor ordinals for β, that is, a sequence of successor ordinals below β with limit β. We first define a family of helper trees $\mathcal{L}_{\beta,k}$ where $k \in \omega \cup \{\infty\}$. Define $\mathcal{L}_{\beta,\infty}$ to consist of a root node whose children are root nodes of copies of $\mathcal{A}_{\beta_i}$, and such that each copy appears exactly once as a child. For $k \in \omega$, $\mathcal{L}_{\beta,k}$ has a root node whose children are root nodes of copies of $\mathcal{A}_{\beta_0}, \ldots, \mathcal{A}_{\beta_k}, \mathcal{E}_{\beta_{k+1}}, \mathcal{E}_{\beta_{k+2}}, \ldots$ where again each copy appears exactly once as a child. Such trees are shown in Fig. 2.4.

We can now define $\mathcal{A}_{\beta+1}$ and $\mathcal{E}_{\beta+1}$ for the non-zero limit ordinal β. For $\mathcal{A}_{\beta+1}$, we have a root node with infinitely many children, each the root node of a copy of $\mathcal{L}_{\beta,k}$ such that for each $k \in \omega$, $\mathcal{L}_{\beta,k}$ appears infinitely many times. The definition of $\mathcal{E}_{\beta+1}$ is similar, except k is drawn from $\omega \cup \{\infty\}$. See Fig. 2.5.

The tree $\mathcal{A}_1$ has rank 0 while the tree $\mathcal{E}_1$ has rank 1. For $\gamma > 0$, $\mathcal{A}_\gamma$ and $\mathcal{E}_\gamma$ have rank γ. The trees $\mathcal{L}^\gamma_k$ and $\mathcal{L}^\gamma_\infty$ both have rank γ.

We also have the following proposition (Proposition 3.2 in [22]):

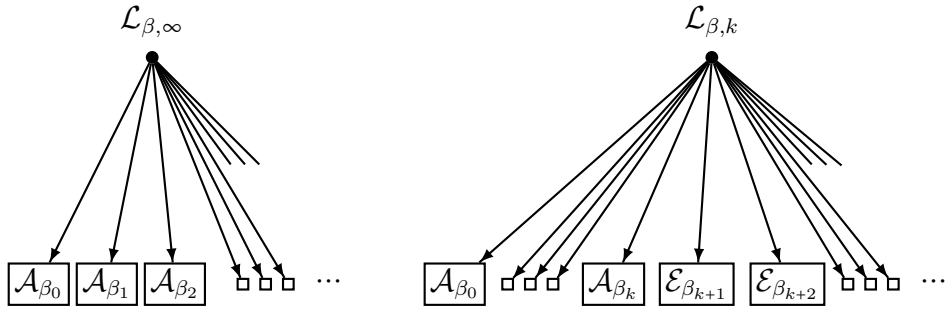


Figure 2.4: Helper trees $\mathcal{L}_{\beta,\infty}$ and $\mathcal{L}_{\beta,k}$ for $k \in \omega$ for the non-zero limit ordinal β.

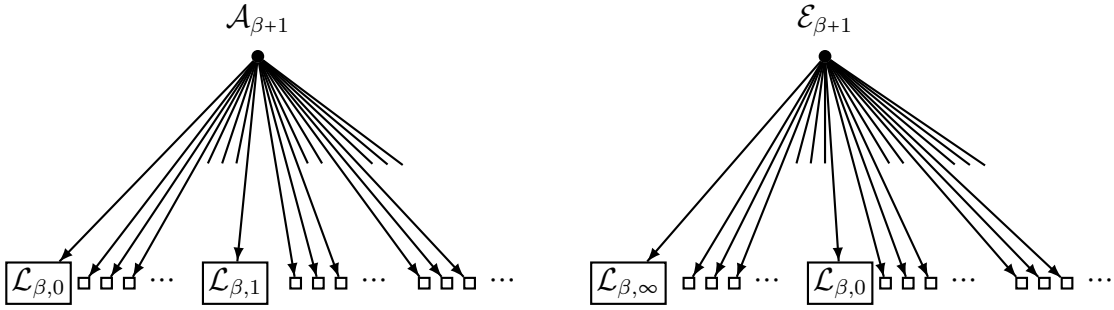


Figure 2.5: $\mathcal{A}_{\beta+1}$ and $\mathcal{E}_{\beta+1}$ for the non-zero limit ordinal β.

Proposition 2.2.6. *Let P be a Σ_α predicate.*

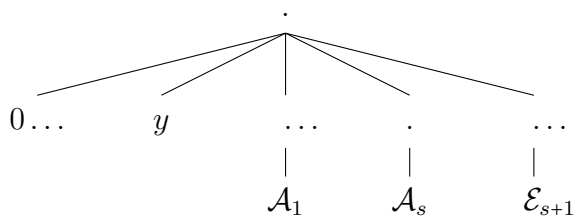
(1) If α is a successor ordinal, there is a sequence of trees $\mathfrak{T}_n$, uniformly computable from a Σ_α index for P such that for all n, $\mathfrak{T}_n \cong \mathcal{E}_\alpha$ if $P(n)$, and $\mathfrak{T}_n \cong \mathcal{A}_\alpha$ if $\neg P(n)$.

(2) If α is a limit ordinal, there is a sequence of trees $\mathfrak{T}_n$, uniformly computable from a Σ_α index for P such that for all n, $\mathfrak{T}_n \cong \mathcal{L}^\alpha_k$ (for some finite k) if $P(n)$, and $\mathfrak{T}_n \cong \mathcal{L}^\alpha_\infty$ if $\neg P(n)$.

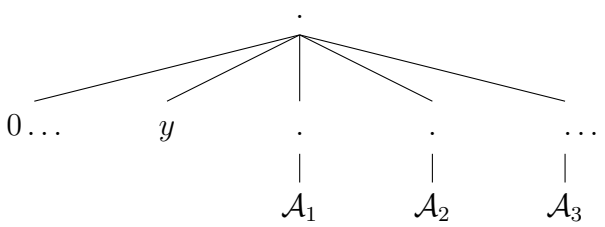
Now we have all the information we need about the trees from [22]. The tree we mentioned earlier to have strong degree of categoricity $\mathbf{0}^{(\omega+1)}$ is in fact isomorphic to $\mathcal{E}_{\omega+1}$.

The next case we consider is trees of rank $\omega + 2$. That case requires us to define new types of trees. Let $B(y, s)$ denote the tree type where: level 1 has exactly $y + 1$ many dead ends, and every other node in level 1 has exactly one immediate successor. Each of these immediate successors (level 2 nodes) is the root of the tree $\mathcal{A}_k$ or $\mathcal{E}_k$ so that they appear as $\mathcal{A}_1, \ldots, \mathcal{A}_s, \mathcal{E}_{s+1}, \mathcal{E}_{s+2} \ldots$. We let $B(y, \omega)$ be the same but the types $\mathcal{E}_k$ do not show up.

$B(y, s)$:



$B(y, \omega)$:



Let $[\omega + 1 : \omega]$ denote the tree type such that:

(1) Every level 1 node is the root of a tree of the type $B(y, s)$ for some $y \in \mathbb{N}$ and $s \in \mathbb{N} \cup \{\omega\}$,

(2) For every natural y, s, the type $B(y, s)$ appears infinitely often.

(3) For every natural y, the type $B(y, \omega)$ appears infinitely often.

Let $[\omega + 1 : m]$ denote the tree type defined exactly like $[\omega + 1 : \omega]$ except that 3 happens exactly for $y \neq m$ (i.e., the type $B(m, \omega)$ does not appear at all, whereas for every $y \neq m$, the type $B(y, \omega)$ apears infinitely often).

Lemma 2.2.7. *Let D be a $\Sigma_{\omega+2}$ set. There is a uniformly computable sequence $(T^x)_{x\in\omega}$ of trees of rank $\omega + 1$ such that $T^x \cong [\omega + 1 : m]$ (for some finite m) if $x \in D$ and $T^x \cong [\omega + 1 : \omega]$ if $x \notin D$.*

Proof. Since D is a $\Sigma_3^{\varnothing^{(\omega)}}$ set, we can apply part 2 of Lemma 2.1.4 to obtain a computable function h such that:

$$x \in D \Leftrightarrow (\exists y)[h(x, y) \in \mathbf{Tot}^{\varnothing^{(\omega)}} \wedge (\forall z \neq y)h(x, z) \in \mathbf{Fin}^{\varnothing^{(\omega)}}],$$

$$x \notin D \Leftrightarrow (\forall y)[h(x, y) \in \mathbf{Fin}^{\varnothing^{(\omega)}}].$$

Let T^x be the tree with root x such that

(1) the level 1 node $(x, 2\langle y, z\rangle)$ has the children $(x, 2\langle y, z\rangle, 0), \ldots, (x, 2\langle y, z\rangle, y)$ as dead ends.

(2) for every natural x, y, z and k, $(x, 2\langle y, z\rangle, y+k+1)$ has exactly one single immediate successor (let it be $(x, 2\langle y, z\rangle, y + k + 1, 0)$),

(3) $(x, 2\langle y, z\rangle, y + k + 1, 0)$ is the root of a tree of type $\mathcal{A}_k$ if $z \notin W^{\varnothing^{(\omega)}}_{h(x,y),k}$, and of type $\mathcal{E}_k$ if $z \in W^{\varnothing^{(\omega)}}_{h(x,y),k}$. This is doable because deciding whether a number enters $W^{\varnothing^{(\omega)}}_{h(x,y)}$ within k stages uses only finitely many pieces from the oracle $\varnothing^{(\omega)}$ (we can assume that the used part is $\varnothing^{(k)}$).

(4) Every node $(x, 2\langle y, z\rangle + 1)$ is the root of a tree of type $B(y, z)$ (this is just to homogenize the tree).

Verification:

Case 1: $x \notin D$. In this case we have that, for every y, $W^{\varnothing^{(\omega)}}_{h(x,y)}$ contains at most finitely many z. This means that, for only finitely many z, z enters $W^{\varnothing^{(\omega)}}_{h(x,y)}$ (at stage s_z, say). Then, $(x, 2\langle y, z\rangle)$ is the root of a tree of type $B(y, s_z)$. For those z that do not enter $W^{\varnothing^{(\omega)}}_{h(x,y)}$ at any stage, $(x, 2\langle y, z\rangle)$ is the root of a tree of type $B(y, \omega)$. So, at the end, $T^x \cong [\omega + 1 : \omega]$.

Case 2: $x \in D$. In this case there exists exactly one y_0 such that $W^{\varnothing^{(\omega)}}_{h(x,y_0)}$ contains every natural number z, and for every $y \neq y_0$, $W^{\varnothing^{(\omega)}}_{h(x,y)}$ contains at most finitely many z. For $y \neq y_0$, we explained in case 1 what the tree rooted at $(x, 2\langle y, z\rangle)$ looks like for every z. Similarly, for every z, one can see $(x, 2\langle y_0, z\rangle)$ is the root of a tree of the type $B(y_0, s_z)$ for some finite s_z. This means that $B(y_0, \omega)$ never shows up. In other words, $T^x \cong [\omega + 1 : y_0]$. □

Theorem 2.2.8. *There is a tree of rank $\omega + 2$ which has strong degree of categoricity $\mathbf{0}^{(\omega+3)}$.*

Proof. Follows from the previous lemma the same way we used Lemma 2.2.2 to prove Theorem 2.2.3, and from Proposition 2.1.2. □

It is easy now to see that we can inductively prove Theorem 2.2.3 for ranks greater than ω using a relativization of Lemma 2.2.2. In what follows we prove a relativization of Lemma 2.2.2, but first some notation: For $n \geq 1$, let $[\omega+n+1 : N]$ denote the tree of rank $\omega+n+1$ consisting of N many trees of the type $[\omega + n : \omega]$ and, for every finite k, infinitely many of the type $[\omega + n : k]$. Also to have the next Lemma written nicely, let $[\omega : \omega]$ denote the type $\mathcal{L}^{\omega}_{\infty}$, and let $[\omega : k]$ denote the type $\mathcal{L}^{\omega}_{k}$.

Lemma 2.2.9. *(Relativization) For every finite $n \geq 0$, given a $\Sigma_{\omega+2n}$ set D, there is a uniformly computable sequence of computable trees $(T^x)_{x\in\omega}$ of rank $\omega + n$ such that for every x*
$x \notin D$ if and only if T^x is of type $[\omega + n : \omega]$ and
$x \in D$ if and only if T^x is of type $[\omega + n : m]$ for some finite m.

Proof. The case $n = 0$ is in Proposition 2.2.6. The case $n = 1$ is Lemma 2.2.7. For $n > 1$, the proof is a raltivization, with respect to $\varnothing^{(\omega)}$, of that of Lemma 2.2.2 (recall that $\Sigma_{\omega+2n} = \Sigma^{\varnothing^{(\omega)}}_{2n+1}$). □

Theorem 2.2.10. *For all $n \geq 0$, there exists a tree of rank $\omega + n + 1$ with strong degree of categoricity $\mathbf{0}^{(\omega+2n+1)}$.*

Proof. Recall that, from Proposition 2.1.2, all we need to do is to show that there is a tree of rank $\omega + n + 1$ with computable copies T_{n+1}, S_{n+1} such that every isomorphism between the two copies computes $\mathbf{0}^{(\omega+2n+1)}$. The case $n = 0$ is the tree $\mathcal{E}_{\omega+1}$ as mentioned in the lines after the proof of Proposition 2.2.6. The case $n > 0$ follows directly from the last lemma by a similar argument to that of Theorem 2.2.3. Let T^x be a computable sequence of trees like in the previous lemma for $D = \varnothing^{(\omega+2n+1)}$ (which is $\Sigma_{\omega+2n}$-complete). Let S_{n+1} be the tree of type $[\omega + n + 1 : \omega]$ with root $()$ and every even level 1 vertex $(2x)$ is the root of T^x and let the odd vertices cover infinitely often the possibilities $[\omega + n : m]$ for all finite m.

Defining T_{n+1}: simply, make every even level 1 node the root of a tree of type $[\omega + n : \omega]$ while keep the odd nodes for infinitely many copies of $[\omega + n : m]$, $m \in \mathbb{N}$.

Now, if $f : S_{n+1} \to T_{n+1}$ is an isomorphism, then for all x, $x \in D$ if and only if $f(2x)$ is odd. □

As the reader can see at the moment, we can easily generalize the previous work to all computable ordinals. First, it is easy to see that Corollary 2.2.4 can be generalized to any rank α that is a computable limit ordinal. This can be done, in the obvious way, by considering a sequence (increasing) of successor ordinals with limit α in place of the natural numbers for ω. For a rank that is a successor of a limit ordinal, such case is handled exactly like the case of rank

$\omega + 1$. Indeed, for a limit ordinal γ, the tree $\mathcal{E}_{\gamma+1}$ has rank $\gamma + 1$ and has strong degree of categoricity $\mathbf{0}^{(\gamma+1)}$ (again, by gluing the disjoint trees composing the structure in Theorem 3.1 in [13]). If the rank is the successor of a successor ordinal, this case is exactly like Theorem 2.2.10. More clearly, we have:

Theorem 2.2.11. *For every computable ordinal α, there exists a computable tree of rank $\alpha+1$ with strong degree of categoricity $\mathbf{0}^{(2\alpha)}$ if α is finite, and with strong degree of categoricity $\mathbf{0}^{(2\alpha+1)}$ if α is infinite. Also, if α is a computable limit ordinal, there exists a computable tree of rank α with strong degree of categoricity $\mathbf{0}^{(\alpha)}$ (= $\mathbf{0}^{(2\alpha)}$).*

Proof. **Case 1:** α is finite. That is exactly Theorem 2.2.3.
Case 2: α is a limit ordinal. Let $(\alpha_k)_{k\in\omega}$ be a computable increasing sequence of successor ordinals with limit α. Assume (induction hypothesis) that for every k, there is a tree of rank $< \alpha$ with two copies $T(k), S(k)$ such that any isomorphism between these two copies computes $\mathbf{0}^{(\alpha_k)}$ (note that this can be immediately deduced from the proof of Theorem 3.1 in [13]). Let S_α be the result of gluing the trees $S(k)$ (as in Corollary 2.2.4), and similarly T_α is the result of gluing the trees $T(k)$ for $k \in \omega$. By Proposition 2.1.2, we have a tree of rank α with strong degree of categoricity $\mathbf{0}^{(\alpha)}$.
Case 3: α is the successor of a limit ordinal γ (i.e. $\alpha = \gamma + 1$). We have the tree $\mathcal{E}_\alpha$.
Case 4: α is the successor of a successor ordinal. We split this case into two subcases:

Subcase 4.1: $\alpha = \gamma + 2$ for some limit ordinal γ. This subcase is exactly like Lemma 2.2.7. All we need to do is to replace the oracle $\varnothing^{(\omega)}$ by $\varnothing^{(\gamma)}$, and the finite portions considered from the oracle will depend on the chosen sequence of successor ordinals which has γ as a limit. Suppose that γ_k is a sequence of successor ordinals with limit γ such that, in order to decide if $z \notin W^{\varnothing^{(\gamma)}}_{h(x,y),k}$, the portion used from the oracle $\varnothing^{(\gamma)}$ is computable from $\varnothing^{(\gamma_k)}$ (think of $\varnothing^{(\gamma)}$ as $\oplus_{k\in\omega}\varnothing^{(\gamma_k)} := \{\langle k, x\rangle : x \in \varnothing^{(\gamma_k)}\}$). Then, in the proof of Lemma 2.2.7, change the following:

(1) In number (3) of the definition of T^x, let $(x, 2\langle y, z\rangle, y + k + 1, 0)$ be the root of a tree of type $\mathcal{A}_{\gamma_k}$ if $z \notin W^{\varnothing^{(\gamma)}}_{h(x,y),k}$ and $\mathcal{E}_{\gamma_k}$ otherwise.

(2) In the definition of $B(y, z)$ (where $z \in \mathbb{N} \cup \{\omega\}$), use $\mathcal{A}_{\beta_k}$ and $\mathcal{E}_{\beta_k}$ in the places of $\mathcal{A}_k$ and $\mathcal{E}_k$.

Subcase 4.2: $\alpha = \gamma + n + 1$ for some $n > 1$. This subcase is exactly like Theorem 2.2.10. Relativizing Lemma 2.2.9 to $\varnothing^{(\gamma)}$ is straightforward. □

Chapter 3

Degrees of Categoricity Above Limit Ordinals

The work in this chapter is joint with Csima, Deveau and Harrison-Trainor (see [12]).

Recall from the introduction that Fokina, Kalimullin, and Miller [15] showed that every degree that can be realized as a difference of computably enumerable (d.c.e.) sets in and above $\mathbf{0}^{(n)}$, for any $n < \omega$, and also the degree $\mathbf{0}^{(\omega)}$, are degrees of categoricity. Later, Csima, Franklin, and Shore [13] showed that every degree $\mathbf{0}^{(\alpha)}$ for every computable ordinal α, and every degree d.c.e. in and above $\mathbf{0}^{(\alpha)}$ for any successor ordinal α, is a degree of categoricity. In this chapter, as we mentioned in the introduction, we fill in a gap above limit ordinals that was missing from the last mentioned work in [13] making a progress towards Question 5.1 of that paper. More precisely, we prove Theorem 3.2.8 below which states that : If α is a computable limit ordinal and $\mathbf{d}$ a degree c.e. in and above $\mathbf{0}^{(\alpha)}$. Then, there is a computable structure with strong degree of categoricity $\mathbf{d}$.

Recall that the *theory* of a structure is the set of first order formulas true in the structure, and that models of the same theory need not be isomorphic. The *type* of a tuple in a structure is the set of formulas (with the appropriate number of free variables) that the tuple satisfies in the structure. The types

of a theory are the types that are realized by models of the theory. A type is called *principal (or isolated)* if there is one formula from which the rest follow (*isolates* the type). A model of a theory is *prime* if it elementarily embeds into all other models of the theory, and when everything is countable, this is the same as saying that the model only realizes principal types (see [30]). In a sense, prime structures are the most basic or natural structures.

Our second result in this chapter gives progress towards a question of Bazhenov and Marchuk.

Question (Bazhenov and Marchuk [5])**.** What can be the degrees of categoricity of computable prime models?

A computable prime model is always $\mathbf{0}^{(\omega+1)}$-categorical (see [5]). Bazhenov and Marchuk constructed a computable homogeneous model (which is implied by being prime for countable models) with degree of categoricty $\mathbf{0}^{(\omega+1)}$. The complexity here is in the structure itself, rather than in the theory. To build a prime model with degree of categoricity $\mathbf{0}^{(\omega+1)}$, the complexity must be in the theory: If $\mathcal{A}$ is a computable prime model of a theory T, then $\mathcal{A}$ is $T' \oplus \mathbf{0}^{(\omega)}$-categorical as T' (the jump of T) can decide whether a formula is complete, and $\mathbf{0}^{(\omega)}$ can decide whether a formula holds of a tuple in $\mathcal{A}$. In Theorem 3.2.10 below, we build a computable prime model with degree of categoricity $\mathbf{0}^{(\omega+1)}$ (or any other degree c.e. in and above $\mathbf{0}^{(\omega)}$), showing that the bound cannot be lowered.

Bazhenov and Marchuk stated in [5] that a careful examination of the structures constructed in [15] shows they are prime models, so that all degrees d.c.e. in and above $\mathbf{0}^{(n)}$ for a finite n, as well as $\mathbf{0}^{(\omega)}$, are strong degrees of categoricity of prime models. Along the way to proving Theorem 3.2.10 we verify in Lemma 3.1.10 that the building blocks used by Csima, Franklin and Shore for their examples in [13] are prime. This is enough to see that their structures realizing degrees of categoricity less than or equal to $\mathbf{0}^{(\omega)}$ are prime. However, the structure in [13] with degree of categoricity $\mathbf{0}^{(\omega+1)}$ is not prime. With Theorem 3.2.10, we see that all known degrees of categoricity less than or equal to the $\mathbf{0}^{(\omega+1)}$ bound can be realized by a prime model.

3.1 Categoricity Relative to Decidable Models

As a warm-up, we give a simple proof of a result of Goncharov [16] that for every c.e. degree $\mathbf{d}$, there is a decidable prime model with degree of categoricity $\mathbf{d}$ with respect to decidable copies. Recall that a structure is said to be decidable if its full elementary diagram is computable. In [16], Goncharov made the following definitions:

Definition 3.1.1. Let $\mathbf{d}$ be a Turing degree and $\mathcal{A}$ a decidable structure. Then $\mathcal{A}$ is $\mathbf{d}$-*categorical with respect to decidable copies* if for every decidable copy $\mathcal{B}$ of $\mathcal{A}$, $\mathbf{d}$ computes an isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Definition 3.1.2. Let $\mathbf{d}$ be a Turing degree and $\mathcal{A}$ a decidable structure. Then $\mathbf{d}$ is the *degree of categoricity of $\mathcal{A}$ with respect to decidable copies* if:

- $\mathcal{A}$ is $\mathbf{d}$-categorical with respect to decidable copies, and
- whenever $\mathcal{A}$ is $\mathbf{c}$-categorical with respect to decidable copies, $\mathbf{c} \geq \mathbf{d}$.

It is not hard to see that between any two decidable copies of a prime model, there is a $\mathbf{0}'$-computable isomorphism. Goncharov showed that any c.e. degree can be the degree of categoricity with respect to decidable copies of a prime model. We give a different proof, which we think is simpler, and which demonstrates some of the techniques that we will use later.

Theorem 3.1.3 (Goncharov [16, Theorem 3])**.** *Let $\mathbf{d}$ be a c.e. degree. Then there is a decidable prime model $\mathcal{M}$ which has strong degree of categoricity $\mathbf{d}$ with respect to decidable models.*

Proof. Let $D \in \mathbf{d}$ be a c.e. set. We will construct the structures $\mathcal{M}$ and $\mathcal{N}$. They are the disjoint union of infinitely many structures $\mathcal{M}_n$ and $\mathcal{N}_n$, with $\mathcal{M}_n$ and $\mathcal{N}_n$ picked out by unary relations R_n. The nth sort will code whether $n \in D$. Fix n. The structure $\mathcal{M}_n$ will have infinitely many elements $(a_i)_{i \in \omega}$. There will be infinitely many unary relations $(U_\ell)_{\ell \in \omega}$ defined on $\mathcal{M}_n$ so that:

$$a_0 \in U_s \iff n \in D_{\text{at } s}$$

where $n \in D_{\text{at } s}$ means that n enters D at exactly stage s ($n \in D_s \setminus D_{s-1}$), and

$$a_i \notin U_s \text{ for } i > 0 \text{ and all } s.$$

Similarly, $\mathcal{N}_n$ will have infinitely many elements $(b_i)_{i \in \omega}$ with the unary relations defined so that:

$$b_i \in U_s \iff i = s \text{ and } n \in D_{\text{at } s}.$$

It is easy to see that we can build computable copies of $\mathcal{M}$ and $\mathcal{N}$. These copies are in fact decidable.

Claim 3.1.4. *$\mathcal{M}$ and $\mathcal{N}$ are decidable.*

Proof. Given a formula $\varphi(x_1, \ldots, x_n)$ with k quantifiers and $a_{i_1}, \ldots, a_{i_n} \in \mathcal{M}$, it is not hard to see that $\mathcal{M} \vDash \varphi(a_{i_1}, \ldots, a_{i_n})$ if and only if the finite substructure $\mathcal{M}'$ of $\mathcal{M}$ whose domain consists of $a_1, \ldots, a_{k+n+1}$ and $a_{i_1}, \ldots, a_{i_n}$ also has $\mathcal{M}' \vDash \varphi(a_{i_1}, \ldots, a_{i_n})$. Thus $\mathcal{M}$ is decidable.

For $\mathcal{N}$, suppose we have a formula $\varphi(x_1, \ldots, x_n)$ with at most k quantifiers and which uses only some subset of the relations $U_0, \ldots, U_k$. Let $b_{i_1}, \ldots, b_{i_n}$ be elements of $\mathcal{N}$. Then $\mathcal{N} \vDash \varphi(b_{i_1}, \ldots, b_{i_n})$ if and only if the finite substructure $\mathcal{N}'$ of $\mathcal{N}$ whose domain consists of $b_1, \ldots, b_{k+n+1}$ and $b_{i_1}, \ldots, b_{i_n}$ also has $\mathcal{N}' \vDash \varphi(b_{i_1}, \ldots, b_{i_n})$. Thus $\mathcal{N}$ is decidable. □

Claim 3.1.5. *$\mathcal{M}$ and $\mathcal{N}$ are isomorphic.*

Proof. It suffices to show that for each n, $\mathcal{M}_n$ and $\mathcal{N}_n$ are isomorphic. If $n \notin D$, then $a_i \mapsto b_i$ induces an isomorphism between $\mathcal{M}_n$ and $\mathcal{N}_n$. If $n \in D_{\text{at } s}$, then the map

$$\begin{aligned} a_0 &\mapsto b_s \\ a_i &\mapsto b_{i-1} && \text{when } 0 < i \leq s \\ a_i &\mapsto b_i && \text{when } i > s \end{aligned}$$

is an isomorphism between $\mathcal{M}_n$ and $\mathcal{N}_n$. □

Claim 3.1.6. *$\mathcal{M}$ and $\mathcal{N}$ are prime.*

Proof. It suffices to show that each $\mathcal{M}_n$ and $\mathcal{N}_n$ are prime, since these structures are determined inside $\mathcal{M}$ and $\mathcal{N}$ uniquely by the relation R_n. It is not hard to see that $\mathcal{M}_n$ and $\mathcal{N}_n$ are models of an $\aleph_0$-categorical theory (a theory such that any two countable models of it are isomorphic), and hence are prime. □

Claim 3.1.7. *Any isomorphism between $\mathcal{M}$ and $\mathcal{N}$ can compute D.*

Proof. Let g be an isomorphism between $\mathcal{M}$ and $\mathcal{N}$. For each n, let $(a_i)_{i\in\omega}$ and $(b_i)_{i\in\omega}$ be the elements in the definition of $\mathcal{M}_n$ and $\mathcal{N}_n$. Let s be such that $g(a_0) = b_s$. Then $n \in D$ if and only if $n \in D_s$. □

Claim 3.1.8. *Given a decidable copy $\widetilde{\mathcal{M}}$ of $\mathcal{M}$, D can compute an isomorphism between $\mathcal{M}$ and $\widetilde{\mathcal{M}}$.*

Proof. For each n, let $\widetilde{\mathcal{M}}_n$ be the structure with domain R_n in $\widetilde{\mathcal{M}}$. It suffices to compute an isomorphism g between $\mathcal{M}_n$ and $\widetilde{\mathcal{M}}_n$ for each n. Let $(c_i)_{i\in\omega}$ be the elements of $\widetilde{\mathcal{M}}_n$. If $n \notin D$, no relation U_j holds of any of the the elements $(a_i)_{i\in\omega}$ or $(c_i)_{i\in\omega}$. So $a_i \mapsto c_i$ is an isomorphism. On the other hand, if $n \in D$, then for some unique s, $a_0 \in U_s$. We can look for c_k such that $c_k \in U_s$. Map a_0 to c_k; map each other a_i to some other c_i. □

These claims complete the proof of the theorem. □

The next two lemmas piece together the facts that we will need about the back-and-forth trees (recall Definition 2.2.5), first for arbitrary β, and second some additional properties for finite β in particular. These facts come from [22] and [13].

Lemma 3.1.9. *Let α be a computable ordinal. For a successor ordinal $\beta < \alpha$, the structures $\mathcal{A}_\beta$ and $\mathcal{E}_\beta$ (from Definition 2.2.5) satisfy the following properties:*

(1) Uniformly in β and an index for a Σ_β set S, there is a computable sequence of structures $\mathcal{C}_x$ such that

$$x \in S \iff \mathcal{C}_x \cong \mathcal{E}_\beta \quad \text{and} \quad x \notin S \iff \mathcal{C}_x \cong \mathcal{A}_\beta.$$

(2) Uniformly in β, there is a Σ^0_β sentence φ such that $\mathcal{E}_\beta \models \varphi$ and $\mathcal{A}_\beta \not\models \varphi$.

(3) $\mathcal{A}_\beta$ and $\mathcal{E}_\beta$ are uniformly $\mathbf{0}^{(\beta)}$-categorical.

Proof. For (1), take $(\mathcal{C}_x)_x$ to be the computable sequence of trees given by Proposition 3.2 in [22].

For (2), take φ to be the sentence given by evaluating the formula guaranteed by Lemma 3.5 in [22] for $\mathcal{B} = \mathcal{E}_\beta$ at its own root node. The complexity of φ is the natural complexity of $\mathcal{E}_\beta$, which is Σ_β. This lemma says that for any tree $\mathcal{T}$, $\mathcal{T} \models \varphi$ if and only if $\mathcal{T} \cong \mathcal{E}_\beta$.

Finally, for (3), we use a result from Csima, Franklin, and Shore [13] about back-and-forth trees. We will consider $\mathcal{A}_\beta$; the case for $\mathcal{E}_\beta$ is identical. We have that $\mathcal{A}_\beta$ is a back-and-forth tree, and hence if $\mathcal{C}$ is a computable structure isomorphic to $\mathcal{A}_\beta$, then it is also a computable back-and-forth tree. Corollary 2.6 in [13] allows $\varnothing^{(\gamma)}$ to uniformly compute an isomorphism between these two trees when the rank of the trees is at most γ. Since the rank of $\mathcal{A}_\beta$ is β by construction, the isomorphism is uniformly computable in $\mathbf{0}^{(\beta)}$. □

Now for finite ordinals β (and writing n for β), we have some additional properties. We will state the lemma in full, including properties that were covered by the previous lemma. We think that these facts are well-known, but we do not know of a reference in print.

Lemma 3.1.10. *For $0 < n < \omega$, the structures $\mathcal{A}_n$ and $\mathcal{E}_n$ satisfy the properties:*

(1) Uniformly in n and an index for a Σ_n set S, there is a computable sequence of structures $\mathcal{C}_x$ such that

$$x \in S \iff \mathcal{C}_x \cong \mathcal{E}_n \quad \text{and} \quad x \notin S \iff \mathcal{C}_x \cong \mathcal{A}_n.$$

(2) For each n, there is an elementary first-order existential sentence φ_n, computable uniformly in n, such that $\mathcal{E}_n \models \varphi_n$ and $\mathcal{A}_n \not\models \varphi_n$.

(3) $\mathcal{A}_n$ and $\mathcal{E}_n$ are prime.

(4) $\mathcal{A}_n$ and $\mathcal{E}_n$ are $\mathbf{0}^{(n)}$-categorical uniformly in n.

Proof. (1) and (4) are the same as in the previous lemma. We show using induction on n that these sequences satisfy (2) and (3) as well. It is easy to see that $\mathcal{A}_1$ and $\mathcal{E}_1$ are prime models of their theories and that they are distinguishable (in the sense of (2) in the statement of the lemma) by the existential sentence $\varphi_1 = \exists x \exists y (x \neq y)$. Assume now that $\mathcal{A}_n$ and $\mathcal{E}_n$ are prime and distinguishable by a first-order existential sentence φ_n (in the sense that $\mathcal{E}_n \models \varphi_n$ but $\mathcal{A}_n \not\models \varphi_n$). We show that $\mathcal{A}_{n+1}$ and $\mathcal{E}_{n+1}$ are prime and distinguishable by a first-order existential sentence φ_{n+1}.

It is not hard to see that we can take φ_{n+1} to be the sentence $\exists x (\neg \varphi_n[\geq x] \wedge$ "x is a child of the root node") where x is a new variable not appearing in φ_n and $\varphi_n[\geq x]$ is the formula obtained from φ_n by bounding every quantifier to the subtree with root x.

It remains to show that $\mathcal{A}_{n+1}$ and $\mathcal{E}_{n+1}$ are prime. We will do that for $\mathcal{E}_{n+1}$ (the same method will work for both structures). Let $\bar{a}$ be an arbitrary tuple in $\mathcal{E}_{n+1}$. We describe a formula that isolates the type of the tuple $\bar{a}$. Let $r_1, \ldots, r_k$ be the level 1 nodes which are the roots of subtrees containing elements of $\bar{a}$. Write $\bar{a} = (\bar{a}_1, \ldots, \bar{a}_k)$ where $\bar{a}_i$ is in the subtree below r_i. Note that we can re-order the tuples as we like (if the type of some permutation of $\bar{a}$ is isolated, then so is that of $\bar{a}$). By the induction hypothesis, we know that the subtree with root r_i is prime for every i; hence, for each $i \in \{1, \ldots, k\}$, there is a formula which isolates the type of the tuple $\bar{a}_i$ in the subtree with root r_i. There is also, for each r_i, a formula (either φ_n or $\neg\varphi_n$) which distinguishes between whether the subtree below r_i is isomorphic to $\mathcal{A}_n$ or $\mathcal{E}_n$. So, we can isolate the type of $\bar{a}$ by saying that there are children $r_1, \ldots, r_k$ of the root such that $\bar{a}_i$ satisfies the formula which isolates it (in the subtree with root r_i), and by saying whether the subtree below each r_i is isomorphic to $\mathcal{A}_n$ or $\mathcal{E}_n$. □

Fokina, Kalimullin, and Miller [15] showed that there is a structure $\mathcal{A}$ with strong degree of categoricity $\mathbf{0}^{(\omega)}$. We note the well-known fact that one can also have $\mathcal{A}$ be a prime model.

Theorem 3.1.11. *There is a computable structure $\mathcal{A}$ with strong degree of categoricity $\mathbf{0}^{(\omega)}$ such that $\mathcal{A}$ is a prime model of its theory.*

Proof. The structure is just the disjoint union of infinitely many copies of

each $\mathcal{E}_n$ for $n < \omega$. Theorem 3.1 of [13] shows that this has strong degree of categoricity $\mathbf{0}^{(\omega)}$, and it is not hard to see using Lemma 3.1.10 that this structure is prime. □

3.2 C.E. In And Above a Limit Ordinal

We begin this section by a short discussion of how we code a c.e. set into a structure. Consider a c.e. set C. If one knows, for each n, at what stage s, $C_s(n) = C(n)$ (i.e. the approximation has settled), then one can compute C. Moreover, one does not need to know exactly when C settles, but just a point after which $C(n)$ has settled. Following the terminology of Groszek and Slaman [18], we say that C has a self-modulus.

Definition 3.2.1 (Groszek and Slaman [18])**.** Let $F:\omega \to \omega$ and $X \subseteq \omega$. Then:

- F is a modulus (of computation) for X if every $G:\omega \to \omega$ that dominates F pointwise computes X.
- X has a self-modulus if X computes a modulus for itself.

The self-modulus of a c.e. set C is the function $f(n) = \mu s(C_s(n) = C(n))$. Groszek and Slaman showed that every Δ_2 set has a self-modulus. In fact, the self-modulus of a c.e. set has a nice form; it has a non-decreasing computable approximation.

Definition 3.2.2. A function $F:\omega \to \omega$ is limitwise monotonic if there is a computable approximation function $f:\omega \times \omega \to \omega$ such that, for all n,

- $F(n) = \lim_{s\to\infty} f(n,s)$.
- For all s, $f(n,s) \leq f(n,s+1)$.

In fact, it is well-known that the sets of c.e. degree are exactly those with limitwise monotonic self-moduli (Theorem 1.1.4).

The next lemma encodes a limitwise monotonic function into the isomorphisms of copies of a computable structure. Any isomorphism dominates the limitwise monotonic function; but it does not seem to be the case that dominating the limitwise monotonic function is sufficient to compute isomorphisms.

Lemma 3.2.3. *Let α be a computable limit ordinal. Let $F:\omega \to \omega$ be limitwise monotonic relative to $\mathbf{0}^{(\alpha)}$ (i.e. the approximation function is computable in $\mathbf{0}^{(\alpha)}$). There is a structure with computable copies $\mathcal{M}$ and $\mathcal{N}$ such that:*

(1) Every isomorphism between $\mathcal{M}$ and $\mathcal{N}$ computes a function which dominates F.

(2) $F \oplus \mathbf{0}^{(\alpha)}$ computes an isomorphism between any two computable copies of $\mathcal{M}$ and $\mathcal{N}$.

Proof. Let Φ be a computable operator such that $F(n) = \lim_{s\to\infty} \Phi^{\varnothing^{(\alpha)}}(n,s)$ and this is monotonic in s. Write $\varnothing^{(\alpha)} = \bigoplus_{\gamma<\alpha} \varnothing^{(\gamma)}$ for successor ordinals $\gamma < \alpha$. By convention, for $\beta < \alpha$, we say that $\Phi^{\varnothing^{(\beta)}}(n,s)$ converges if the computation $\Phi^{\varnothing^{(\alpha)}}(n,s)$ halts, but the only part of the oracle $\varnothing^{(\alpha)} = \bigoplus_{\gamma<\alpha} \varnothing^{(\gamma)}$ that is read during the computation is that part with $\gamma \leq \beta$. So, if $\Phi^{\varnothing^{(\beta)}}(n,s) = m$, then $\Phi^{\varnothing^{(\alpha)}}(n,s) = m$. Since α is a limit ordinal, if $\Phi^{\varnothing^{(\alpha)}}(n,s) = m$, then $\Phi^{\varnothing^{(\beta)}}(n,s) = m$ for some successor ordinal $\beta < \alpha$.

Let $(\mathcal{A}_\beta)_{\beta<\alpha}$ and $(\mathcal{E}_\beta)_{\beta<\alpha}$ be as in Lemma 3.1.9. We will construct the structures $\mathcal{M}$ and $\mathcal{N}$. They are the disjoint union of infinitely many structures $\mathcal{M}_n$ and $\mathcal{N}_n$, with $\mathcal{M}_n$ and $\mathcal{N}_n$ picked out by unary relations R_n. The nth sort will code the value of $F(n)$.

Fix n. $\mathcal{M}_n$ will have infinitely many elements $(a_i)_{i\in\omega}$ satisfying a unary relation S. Each of these elements will be attached to, for each successor ordinal $\beta < \alpha$, a "box" $\mathcal{M}_{i,\beta}$ which contains within it a copy of either $\mathcal{A}_\beta$ or $\mathcal{E}_\beta$; each of the boxes are disjoint. By this we mean that there are binary relations T_β such that $T_\beta(a_i, x)$ holds for exactly those $x \in \mathcal{M}_{i,\beta}$. The structure $\mathcal{M}_{i,\beta}$ will be a structure in the language of Lemma 3.1.9 and will be defined so that:

(1) $\mathcal{M}_{0,\beta} \cong \mathcal{A}_\beta$ for all β.

(2) $\mathcal{M}_{i,\beta} \cong \mathcal{E}_\beta$, $i \geq 1$, if there is s such that $\Phi^{\varnothing^{(\beta)}}(n,s) \geq i$.

(3) $\mathcal{M}_{i,\beta} \cong \mathcal{A}_\beta$, $i \geq 1$, otherwise.

Note that the condition in (2) is Σ_β and so we can build such a structure $\mathcal{M}_n$ computably.

Similarly, $\mathcal{N}_n$ will have infinitely many elements $(b_i)_{i\in\omega}$, each of which is attached to, for each $\beta < \alpha$, a box $\mathcal{N}_{i,\beta}$ which contains within it:

(1) $\mathcal{N}_{i,\beta} \cong \mathcal{E}_\beta$ if there is s such that $\Phi^{\varnothing^{(\beta)}}(n,s) > i$.

(2) $\mathcal{N}_{i,\beta} \cong \mathcal{A}_\beta$ otherwise.

Again, the condition in (1) is Σ_β and so we can build such a structure $\mathcal{N}_n$ computably.

Claim 3.2.4. *Fix n.*

(1) For each $j < F(n)$, there is $\beta < \alpha$ such that:

- *for $\gamma < \beta$, $\mathcal{M}_{j+1,\gamma} \cong \mathcal{N}_{j,\gamma} \cong \mathcal{A}_\gamma$,*
- *for $\gamma \geq \beta$, $\mathcal{M}_{j+1,\gamma} \cong \mathcal{N}_{j,\gamma} \cong \mathcal{E}_\gamma$,*

(2) For each $j \geq F(n)$ and $\beta < \alpha$, $\mathcal{M}_{j+1,\beta} \cong \mathcal{N}_{j,\beta} \cong \mathcal{M}_{0,\beta} \cong \mathcal{A}_\beta$.

Proof. For (1), it is clear from the definitions of $\mathcal{M}_{j+1,\beta}$ and $\mathcal{N}_{j,\beta}$ that for all $\beta < \alpha$, $\mathcal{M}_{j+1,\beta} \cong \mathcal{N}_{j,\beta}$. Since $j < F(n)$, there is s such that $\Phi^{\varnothing^{(\alpha)}}(n,s) = F(n) > j$. In particular, there must be some $\beta < \alpha$ such that there is s with $\Phi^{\varnothing^{(\beta)}}(n,s) > j$. Let β be the least such ordinal. Then for all $\gamma \geq \beta$, there is s such that $\Phi^{\varnothing^{(\beta)}}(n,s) > j$, and so $\mathcal{M}_{j+1,\gamma} \cong \mathcal{N}_{j,\gamma} \cong \mathcal{E}_\gamma$. By choice of β, for $\gamma < \beta$, there is no s such that $\Phi^{\varnothing^{(\beta)}}(n,s) > j$, and so $\mathcal{M}_{j+1,\gamma} \cong \mathcal{N}_{j,\gamma} \cong \mathcal{A}_\gamma$.

For (2), it is clear that $\mathcal{M}_{j+1,\beta} \cong \mathcal{N}_{j,\beta}$ for each $j \geq F(n)$ and each $\beta < \alpha$, and it is also clear that $\mathcal{M}_{0,\beta} \cong \mathcal{A}_\beta$ for each $\beta < \alpha$. If $j \geq F(n)$, then since Φ is limitwise monotonic approximation to F, $\Phi^{\varnothing^{(\beta)}}(n,s) \leq F(n) \leq j$ for all s and β. Thus $\mathcal{N}_{j,\beta} \cong \mathcal{A}_\beta$ for all β. □

Claim 3.2.5. *$\mathcal{M}$ and $\mathcal{N}$ are isomorphic.*

Proof. It suffices to show that for each n, $\mathcal{M}_n$ and $\mathcal{N}_n$ are isomorphic. Fix n. Using Claim 3.2.4, we see that the map

$$\begin{aligned} &a_0 \mapsto b_{F(n)} && \\ &a_i \mapsto b_{i-1} && \text{when } 0 < i \leq F(n) \\ &a_i \mapsto b_i && \text{when } i > F(n) \end{aligned}$$

extends to an isomorphism between $\mathcal{M}_n$ and $\mathcal{N}_n$. □

Claim 3.2.6. *Any isomorphism between $\mathcal{M}$ and $\mathcal{N}$ can compute a function which dominates F.*

Proof. Let g be an isomorphism between $\mathcal{M}$ and $\mathcal{N}$. We will compute, using g, a function $\hat{g}$ which dominates F. For each n, define $\hat{g}(n)$ as follows. Let $(a_i)_{i\in\omega}$ and $(b_i)_{i\in\omega}$ be the elements in the definition of $\mathcal{M}_n$ and $\mathcal{N}_n$. Then $\hat{g}(n)$ is the number satisfying $g(a_0) = b_{\hat{g}(n)}$.

We have that $\hat{g}(n) \geq F(n)$. For each $\beta < \alpha$, $\mathcal{M}_{0,\beta} \cong \mathcal{A}_\beta$, but if $j < F(n)$, there is $\beta < \alpha$ such that $\mathcal{N}_{j,\beta} \cong \mathcal{E}_\beta$ (by Claim Theorem 3.2.4). Thus, no isomorphism can map a_0 to b_j for $j < F(n)$, and so $\hat{g}(n) \geq F(n)$. □

Claim 3.2.7. *Given a computable copy $\widetilde{\mathcal{N}}$ of $\mathcal{N}$, $F \oplus \mathbf{0}^{(\alpha)}$ can compute an isomorphism between $\mathcal{N}$ and $\widetilde{\mathcal{N}}$.*

It is more convenient for the proof to consider $\mathcal{N}$ rather than $\mathcal{M}$ in this claim, but as they are isomorphic it does not matter which we choose.

Proof. For each n, let $\widetilde{\mathcal{N}}_n$ be the structure with domain R_n in $\widetilde{\mathcal{N}}$. It suffices to compute an isomorphism g between $\mathcal{N}_n$ and $\widetilde{\mathcal{N}}_n$ for each n. Inside of $\widetilde{\mathcal{N}}_n$, let $(c_i)_{i\in\omega}$ list the elements x satisfying $S(x)$. For each c_i, let $\widetilde{\mathcal{N}}_{i,\beta}$ be the tree whose domain consists of the elements y satisfying $T_\beta(c_i, y)$. To begin, we will define g on $(b_i)_{i\in\omega} \subseteq \mathcal{N}_n$. Compute $F(n)$. Using $\mathbf{0}^{(\alpha)}$, look for $F(n)$ elements c_i such that, for some $\beta < \alpha$, $\widetilde{\mathcal{N}}_{i,\beta} \cong \mathcal{E}_\beta$. This search is computable relative to $\mathbf{0}^{(\alpha)}$ by Lemma 3.1.9 (2), and by Claim 3.2.4 we know that there are exactly $F(n)$ such elements and so the search will terminate after finding every such element. Rearranging $(c_i)_{i\in\omega}$, we may assume that these elements are $c_0, \ldots, c_{F(n)-1}$.

Now, for each $k < F(n)$, find the least β_k such that $\mathcal{N}_{k,\beta_k} \cong \mathcal{E}_{\beta_k}$, and the least γ_k such that $\widetilde{\mathcal{N}}_{k,\gamma_k} \cong \mathcal{E}_{\gamma_k}$. Again, this is computable in $\mathbf{0}^{(\alpha)}$ by Lemma 3.1.9 (2). Note that we must ask $\mathbf{0}^{(\alpha)}$ to determine what β_k and γ_k are least. The sets $\{\beta_0, \ldots, \beta_{F(n)-1}\}$ and $\{\gamma_0, \ldots, \gamma_{F(n)-1}\}$ must be identical including multiplicity (but possibly in a different order) as $\widetilde{\mathcal{N}}_n$ and $\mathcal{N}_n$ are isomorphic. So by rearranging $(c_i)_{i\in\omega}$ once again we may assume that $\beta_k = \gamma_k$ for each $k < F(n)$.

We have now rearranged the list $(c_i)_{i\in\omega}$ so that for each i and $\beta < \alpha$, $\mathcal{N}_{i,\beta} \cong \widetilde{\mathcal{N}}_{i,\beta}$. Define g so that $g(a_i) = c_i$. For each i and $\beta < \alpha$, $\mathcal{N}_{i,\beta} \cong \widetilde{\mathcal{N}}_{i,\beta}$ are isomorphic to either $\mathcal{A}_\beta$ or $\mathcal{E}_\beta$, which are uniformly $\mathbf{0}^{(\beta)}$-categorical (Lemma 3.1.9 (3)), and we can compute using $\mathbf{0}^{(\alpha)}$ which case we are in. So we can define g on $\mathcal{N}_{i,\beta}$ to be an isomorphism to $\widetilde{\mathcal{N}}_{i,\beta}$. Thus g is an isomorphism from $\mathcal{N}_n$ to $\widetilde{\mathcal{N}}_n$. □

These claims complete the proof of the theorem. □

Using this lemma, and taking the limitwise monotonic function to be the self-modulus of a c.e. set, it is not hard to prove our main theorem.

Theorem 3.2.8. *Let α be a computable limit ordinal and $\mathbf{d}$ a degree c.e. in and above $\mathbf{0}^{(\alpha)}$. There is a computable structure with strong degree of categoricity $\mathbf{d}$.*

Proof. Fix α and let $D \in \mathbf{d}$ be a set c.e. in and above $\mathbf{0}^{(\alpha)}$. Since D is c.e. in and above $\mathbf{0}^{(\alpha)}$, it has a self-modulus f that is limitwise monotonic relative to $\mathbf{0}^{(\alpha)}$. Consider the structure $\mathcal{M}$ constructed in Lemma 3.2.3 for this f. We will enrich this structure slightly to produce a new structure $\mathcal{S}$. Let $\mathcal{S}_\alpha$ be the computable structure with strong degree of categoricity $\mathbf{0}^{(\alpha)}$ constructed in Theorem 3.1 of Csima, Franklin and Shore [13]. The new structure $\mathcal{S}$ consists of $\mathcal{M}$ and a disjoint copy of $\mathcal{S}_\alpha$, and a new unary relation R such that $R(x)$ holds exactly when x belongs to the copy of $\mathcal{S}_\alpha$. We claim that $\mathcal{S}$ has strong degree of categoricity $\mathbf{d}$.

First, suppose that $\mathcal{T}$ is some other computable copy of $\mathcal{S}$. We will show that there is a $\mathbf{d}$-computable isomorphism between $\mathcal{S}$ and $\mathcal{T}$. Using the relation R, we may identify the component of $\mathcal{T}$ isomorphic to $\mathcal{S}_\alpha$. Since $\mathcal{S}_\alpha$ has strong degree of categoricity $\mathbf{0}^{(\alpha)} \leq \mathbf{d}$, we can $\mathbf{d}$-computably find an isomorphism between the copies of $\mathcal{S}_\alpha$ in $\mathcal{S}$ and $\mathcal{T}$. We can also identify the component isomorphic to $\mathcal{M}$ in each structure. By choice of $\mathcal{M}$, any two such copies have an isomorphism between them computable in $f \oplus \mathbf{0}^{(\alpha)}$, and D can compute this self-modulus f. Hence $\mathbf{d}$ can computably produce such an isomorphism, since it can compute $f \oplus \mathbf{0}^{(\alpha)}$. Gluing these two isomorphisms together gives us the result.

Since $\mathcal{S}_\alpha$ has strong degree of categoricity $\mathbf{0}^{(\alpha)}$, there is a computable copy $\hat{\mathcal{S}}_\alpha$ of $\mathcal{S}_\alpha$ such that every isomorphism between the two computes $\mathbf{0}^{(\alpha)}$. Let $\widetilde{\mathcal{S}}$ be a computable copy of $\mathcal{S}$ built in the following way. Rather than using the "standard" copy $\mathcal{S}_\alpha$, use the "hard" copy $\hat{\mathcal{S}}_\alpha$ of $\mathcal{S}_\alpha$. Additionally, rather than using $\mathcal{M}$, instead use $\mathcal{N}$ as built in Lemma 3.2.3. Any isomorphism between $\mathcal{S}_\alpha$ and $\hat{\mathcal{S}}_\alpha$ computes $\mathbf{0}^{(\alpha)}$, and any isomorphism between $\mathcal{M}$ and $\mathcal{N}$ must compute a function that dominates f. Let g be any isomorphism between $\mathcal{S}$ and $\widetilde{\mathcal{S}}$. Then by using R, we can restrict g to an isomorphism between $\mathcal{S}_\alpha$ and $\hat{\mathcal{S}}_\alpha$ and hence g can compute $\mathbf{0}^{(\alpha)}$. Since g can also be restricted to an isomorphism between $\mathcal{M}$ and $\mathcal{N}$, it must compute a function dominating F. But F is a modulus for D computable in $\mathbf{0}^{(\alpha)}$, and hence g must be able to compute D since it can compute $\mathbf{0}^{(\alpha)}$ and a function dominating f. Hence g can compute $\mathbf{d}$. □

We now turn to prime models, working above $\mathbf{0}^{(\omega)}$. Essentially, our work here is to check that in taking $\alpha = \omega$ in the previous theorem and lemma, the construction results in a prime model.

Lemma 3.2.9. *Let $f:\omega \to \omega$ be limitwise monotonic relative to $\mathbf{0}^{(\omega)}$. There is a prime model with two computable copies $\mathcal{M}$ and $\mathcal{N}$ such that:*

(1) Every isomorphism between $\mathcal{M}$ and $\mathcal{N}$ computes a function which dominates F.

(2) $F \oplus \mathbf{0}^{(\omega)}$ computes an isomorphism between any two computable copies of $\mathcal{M}$ and $\mathcal{N}$.

Proof. The construction is exactly the same as that of Lemma 3.2.3 with $\alpha = \omega$. We refer to the structures $\mathcal{A}_\beta$ and $\mathcal{E}_\beta$ of Theorem 3.1.9 as $\mathcal{A}_n$ and $\mathcal{E}_n$, $n < \omega$, but of course these are the same. It remains to argue, using the properties from Lemma 3.1.10 which hold only for the structures $\mathcal{A}_n$ and $\mathcal{E}_n$ with n finite, that the resulting structure $\mathcal{N}$ is prime.

Recall that $\mathcal{N}$ is the disjoint union of structures $\mathcal{N}_n$, each of which satisfies the relation R_n. So it suffices to show that the structures $\mathcal{N}_n$ are prime. $\mathcal{N}_n$ was defined as follows: there were infinitely many elements $(b_i)_{i\in\omega}$ (satisfying

the unary relation S), each of which is attached to (by binary relations T_m), for each $m < \omega$, a box $\mathcal{N}_{i,m}$ which contains within it:

(1) $\mathcal{N}_{i,m} \cong \mathcal{E}_m$ if there is s such that $\Phi^{\varnothing^{(m)}}(n, s) > i$.

(2) $\mathcal{N}_{i,m} \cong \mathcal{A}_m$ otherwise.

By Theorem 3.2.4 of Theorem 3.2.3, for each i, either $i < F(n)$ and there is some $m_i < \omega$ such that:

- for $\ell < m_i$, $\mathcal{N}_{i,\ell} \cong \mathcal{A}_\ell$,
- for $\ell \geq m_i$, $\mathcal{N}_{i,\ell} \cong \mathcal{E}_\ell$,

or $i \geq F(n)$ and for all $m < \omega$, $\mathcal{N}_{i,m} \cong \mathcal{A}_m$. Note that the sequence $\{m_i\}_{i<F(n)}$ is non-decreasing.

By Lemma 3.1.10 (2), for $i < F(n)$, the automorphism orbit of b_i is determined by the first-order formula with free variable x which expresses that S holds of x, that the structure with domain $T_{m_i}(x, \cdot)$ satisfies φ_{m_i} (and so is isomorphic to $\mathcal{E}_{m_i}$), and that the structure with domain $T_{m_i-1}(x, \cdot)$ satisfies $\neg\varphi_{m_i-1}$ (and so is isomorphic to $\mathcal{A}_{m_i-1}$). For $i \geq F(n)$, the automorphism orbit of b_i is determined by the first-order sentence with free variable x which expresses that S holds of x, and that the structure with domain $T_{m_{F(n)-1}}(x, \cdot)$ satisfies $\neg\varphi_{m_{F(n)-1}}$ (and so is isomorphic to $\mathcal{A}_{m_{F(n)-1}}$).

Fix a tuple $\bar{c}$ from $\mathcal{N}_n$. We will give a first-order formula defining the orbit of $\bar{c}$. We may assume that whenever $\bar{c}$ contains an element of $\mathcal{N}_{i,m}$, $\bar{c}$ contains b_i as well. We can break the tuple $\bar{c}$ up into finitely many elements $b_{i_1}, \ldots, b_{i_k}$ and finitely many tuples $\bar{c}_{i,m}$ from $\mathcal{N}_{i,m}$. The orbit of $\bar{c}$ is determined by the orbits of $b_{i_1}, \ldots, b_{i_k}$ (each of which is determined by a first-order formula as described in the previous paragraph), the fact that $T_m(b_i, y)$ holds for any $y \in \bar{c}_{i,m}$, and the orbits of each of the tuples $\bar{c}_{i,m}$ within $\mathcal{N}_{i,m}$. The latter orbits are first-order definable by Theorem 3.1.10 (3). □

Theorem 3.2.10. *Let* $\mathbf{d}$ *be a degree c.e. in and above* $\mathbf{0}^{(\omega)}$. *There is a computable prime model* $\mathcal{A}$ *with strong degree of categoricity* $\mathbf{d}$.

Proof. The construction of such a model is similar to Theorem 3.2.8, except we replace $\mathcal{M}$ and $\mathcal{N}$ from Theorem 3.2.3 with those $\mathcal{M}$ and $\mathcal{N}$ from Lemma 3.2.9 (which are actually the same structures, if $\alpha = \omega$), and we also replace the "easy" and "hard" copies of $\mathcal{S}_\alpha$ with copies of the structure from Theorem 3.1.11 such that any isomorphism between them computes $\mathbf{0}^{(\omega)}$. The same argument from Theorem 3.2.8 shows that this new structure has strong degree of categoricity $\mathbf{d}$. It remains to show that such models are prime. Indeed, they are the disjoint union of prime structures, distinguishable by the relation R, and hence must be prime themselves. □

Chapter 4

The Isomorphism Problem

In this chapter, we are interested in knowing how hard it is to decide for two given computable structures (in certain classes) whether they are isomorphic or non-isomorphic.

Recall the following definition from the introduction:

Definition 4.0.1 (Goncharov and Knight [17])**.** The *isomorphism problem*, denoted $I(K)$, for a class K of computable structures, is the set $I(K) = \{(n, m) : \mathcal{M}_n, \mathcal{M}_m \in K, \mathcal{M}_n \cong \mathcal{M}_m\}$ where $(\mathcal{M}_n)_{n\in\omega}$ is some computable enumeration of all computable structures (on a fixed computable language).

4.1 The Isomorphism Problem for Trees

In Chapter 2, we studied the degrees of categoricity for tree structures. It is immediate to notice a connection between Definition 4.0.1 and the work in Chapter 2. Recall that Lemma 2.1.1 proves that checking whether two computable trees of rank α are isomorphic is $\Pi_{2\alpha}$-uniformly. This implies that, for $\alpha > 0$, if K_α is the class of computable trees of rank α, then $I(K_\alpha)$ is $\Pi_{2\alpha}$. We dropped the case $\alpha = 0$ because deciding membership for the class of computable trees is Π_2. Note that deciding whether a computable tree is of rank $\alpha > 0$ can be done within $\Pi_{2\alpha}$ for all computable ordinals α. This can be proved by an inductive argument.

In this section, we use our work in Chapter 2 to show that $I(K_\alpha)$ is in fact

$\Pi_{2\alpha}$-complete. Recall that a set X in a class Γ is said to be Γ-complete if every set in Γ is m-reducible (equivalently, 1-reducible) to X.

Theorem 4.1.1. *For every computable ordinal $\alpha > 0$, the isomorphism problem for the class of computable trees of rank α is $\Pi_{2\alpha}$-complete.*

In light of Lemma 2.1.1, the proof will be a one-liner once we introduce some notation and a lemma. For a limit ordinal α, let $[\alpha : m]$ and $[\alpha : \omega]$ denote the trees $\mathcal{L}^{\alpha}_{m}$ and $\mathcal{L}^{\alpha}_{\infty}$ from [22], respectively. If α is the successor of a limit ordinal, say $\alpha = \gamma + 1$, we define $[\alpha : m]$ and $[\alpha : \omega]$ the same way we defined $[\omega + 1 : m]$ and $[\omega + 1 : \omega]$. We just replace $\mathcal{A}_k$ by $\mathcal{A}_{\gamma_k}$ and $\mathcal{E}_k$ by $\mathcal{E}_{\gamma_k}$ where $(\gamma_k)_{k\in\omega}$ is a computable increasing sequence of successor ordinals with limit γ. If α is the successor of a successor, then $[\alpha : m]$ and $[\alpha : \omega]$ can be defined the obvious way inductively. This is exactly as how we defined $[\omega + n + 1 : N]$ before. For $n \geq 1$, let $[\alpha + n + 1 : N]$ denote the tree of rank $\alpha + n + 1$ consisting of N (can be ω) many trees of the type $[\alpha + n : \omega]$ and, for every finite k, infinitely many of the type $[\alpha + n : k]$.

Lemma 4.1.2. *Let α be a computable ordinal and let C be a $\Pi_{2\alpha}$ set. There is a sequence $(T^x)_{x\in\omega}$ of trees of rank α uniformly computable from an index of C such that T^x is of the type $[\alpha : \omega]$ if $x \in C$, and of the type $[\alpha : m]$ for some finite m if $x \notin C$.*

Proof. The case $\alpha = 0$ is trivial. The case when $\alpha > 0$ is finite follows immediately from Lemma 2.2.2 (apply it to the complement of C). The case when α is infinite is an easy generalization of Lemma 2.2.9 which is used in the proof of Theorem 2.2.11. □

Proof of Theorem 4.1.1. Fix α. Let C and T^x be as in Lemma 4.1.2. Consider the following function $\eta : \omega \to \omega$ given by $\eta(x) = \langle \ulcorner T^x \urcorner, \ulcorner [\alpha : \omega] \urcorner \rangle$ where the $\ulcorner . \urcorner$ indicates the number of the structure according to some fixed computable enumeration of computable $\prec$-structures. Clearly η is computable and it is a many-one reduction of C to $I(K_\alpha)$ where K_α is the class of computable trees of rank α. □

It is possibly a little surprising to the reader that Theorem 4.1.1 does not need distinct formulas for finite and infinite α, whereas in Chapter 2 we had to

distinguish between these cases. This makes perfect sense because, for a set, being a $\mathbf{0}^{(\alpha)}$-c.e. is equivalent to being $\Sigma_{\alpha+1}$ when α is finite, but equivalent to being Σ_α when α is infinite.

Remark 4.1.3. It is worth mentioning that a result similar to Theorem 4.1.1 is true for automatic trees. In [26], Kuske, Liu and Lohrey proved that the isomorphism problem for automatic trees of height $n \geq 2$ (natural number) is Π_{2n-3}-complete.

4.2 The Isomorphism Problem for Pregeometries

The work in this section is motivated by that of Harrison-Trainor, Melnikov and Montalbán in [20]. There they gave a sufficient condition for a structure with some notion of independence to have a computable presentation with a computable basis ("good" copy) and another computable presentation with no computable basis ("bad" copy). They applied the condition to differentially closed, real closed, and difference closed fields. The condition also implied classical results on vector spaces, algebraically closed fields, torsion-free groups and Archimedean ordered abelian groups.

In [8], Calvert studied classes that admit a notion of independence. In fact, they are among the types of classes for which the ideas in [20] are applicable. We implement the unified framework built in [20] to generalize some of the work in [8] and [9]. We also take the opportunity to clarify the key methods in the proofs in [8] and to explain the details that were not quite clear.

4.2.1 Preliminaries

The standard abstraction for independence is the notion of a *pregeometry*.

Definition 4.2.1 (See [30])**.** Let X be a set and let $cl : \mathcal{P}(X) \to \mathcal{P}(X)$ be an operator on the power set of X. We say that (X, cl) is a pregeometry if

(1) If $A \subseteq X$, then $A \subseteq cl(A)$ and $cl(cl(A)) = cl(A)$,

(2) if $A \subseteq B \subseteq X$, then $cl(A) \subseteq cl(B)$,

(3) (exchange principle) if $A \subseteq X$, $a, b \in X$, $a \in cl(A \cup \{b\})$, and $a \notin cl(A)$, then $b \in cl(A \cup \{a\})$,

(4) (finite character) if $A \subseteq X$ and $a \in cl(A)$, then there is a finite $A_0 \subseteq A$ such that $a \in cl(A_0)$.

An operator which satisfies the first two properties is called a *closure operator*. Let (X, cl) be a pregeometry, and $A \subseteq X$. We say that

(1) A is *closed* if $A = cl(A)$;

(2) A *spans* a set $B \supseteq X$ if $B \subseteq cl(A)$;

(3) $A \subseteq X$ is *independent* if for all $a \in A$, $a \notin cl(A \setminus \{a\})$, and A is *dependent* otherwise;

(4) B is a basis for $Y \subseteq X$ if B spans Y and is independent.

One can show that B is a basis for Y if, and only if, B is a maximal independent set contained in Y. A standard argument shows that every set has a basis, and that every basis for Y has the same cardinality which is called the *dimension* of Y (see for example [37]).

Because a pregeometry has finite character, all the information of the pregeometry is captured in the relations (for all natural n) $x \in cl(\{y_1, \ldots, y_n\})$. For convenience, we will write $cl(y_1, \ldots, y_n)$ to mean $cl(\{y_1, \ldots, y_n\})$ and will refer to $\{(x, y_1, \ldots, y_n) : x \in cl(_1, \ldots, y_n), n \in \mathbb{N}\}$ as the *pregeometry relation*.

As in [20], we will be working in the context of pregeometries that can be effectively enumerated. We will say cl is c.e. (computably enumerable) if each of the relations $x \in cl(y_1, \ldots, y_n)$ is c.e. uniformly in n. Since we are interested in pregeometries defined on structures and their copies, we have the following definition.

Definition 4.2.2. [20] A pregeometry operator cl on a structure $\mathcal{M}$ is r.i.c.e. (relatively intrinsically computably enumerable) if, uniformly in n, for every fixed n, the relation $x \in cl(y_1, \ldots, y_n)$ is c.e. relative to every copy of $\mathcal{M}$.

Being r.i.c.e. is a general property that can be defined by relations that are not necessarily a pregeometry. In [2, 11], it was shown that a relation is r.i.c.e. if and only if it is definable by an infinitary computable Σ_1 formula. Throughout this section we will use the word "pregeometry" to mean a structure in some first order language (not merely a set) with a pregeometry relation as explained earlier. All standard examples of pregeometries (vector spaces, fields, etc.) are r.i.c.e. pregeometries.

In this section, we will prove the Π_3-hardness of the Isomorphism Problem for classes of pregeometries in which there is an abundance of dependent elements. To understand the last sentence, we recall here the following definitions.

Definition 4.2.3. (Condition B) [20]. In a pregeometry $(\mathcal{M}, cl)$, we say that *dependent elements are dense* if, whenever ψ is quantifier-free and $\mathcal{M} \models \exists \bar{y}\psi(\bar{c}, \bar{y}, a)$ for a non-empty tuple $\bar{c}$ and $a \in \mathcal{M}$, there is $b \in cl(\bar{c})$ and $\mathcal{M} \models \exists y\psi(\bar{c}, \bar{y}, b)$.

It is not hard to see that, for any class of structures K, the hardest $I(K)$ can be is Σ^1_1. Several classes of computable structures are known to have the hardest isomorphism problems. Indeed, for Undirected Graphs, Linear Orders, Trees, Boolean algebras, and Abelian p-groups the isomorphism problem is Σ^1_1-complete. Proofs may be found in articles by Goncharov and Knight [17], Morozov [33], and Nies [34]. Also, the isomorphism problem is Σ^1_1-complete for Rings, Distributive Lattices, Nilpotent Groups, and Semigroups. This follows from the work of Hirschfeldt, Khoussainov, Shore, and Slinko in [21].

In [8], the focus was on calculating the complexity of the isomorphism problem for classes of fields. There Calvert proved that the class of real closed fields can be added to the lists above. However, in the same paper, Calvert also proved that the isomorphism problem is simple for some classes.

Theorem 4.2.4 ([8])**.** *If K is any of the following classes, then $I(K)$ is Π_3-complete.*

(1) Computable vector spaces over a fixed infinite computable field.

(2) Computable algebraically closed fields of fixed characteristic.

(3) Archimedean real closed fields.

It is easy to see that, for any of the last three classes, the isomorphism problem is Π_3 (for example, two vector spaces are isomorphic iff they have the same dimension). Proving the hardness (hence, completeness) for 1 and 2 was established by defining a 1-reduction of the set of indices of coinfinite c.e. sets $\overline{\mathbf{Cof}}$ into $I(K)$. For Archimedian real closed fields, the reduction was of an arbitrary Π_3 set. The key property used in the three proofs for the hardness part is, in essence, Condition B. This is not mentioned directly in [8], but instead the existence of dependent elements is established through strong minimality (for 1,2), or through some basic topological argument for 3. Applying Condition B directly makes the arguments clearer and allows us to prove the Π_3-hardness for other classes.

The following are more examples of classes for which Condition B holds and for which the isomorphism problem hasn't been considered before.

(1) Algebraically closed valued fields. By the proof of Theorem 5.6 in [19], an algebraically closed valued field of infinite transcendence degree satisfies Condition B.

(2) The class of p-adically closed fields. By the proof of Theorem 5.8 in [19], a p-adically closed field of infinite transcendence degree satisfies Condition B.

4.2.2 When Dependent Elements are Dense

In this section, we prove a general theorem (Theorem 4.2.6 below) that captures the results mentioned in the earlier sections. We assume our structures are for some computable infinite language with equality. Our proofs work for finite languages as well.

First, let us start by rephrasing Condition B in a more helpful way.

Lemma 4.2.5. *Condition B is equivalent to the following: For every quantifier-free formula ψ and non-empty tuples $\bar{a}, \bar{c}$ in $\mathcal{M}$, if $\mathcal{M} \vDash \psi(\bar{c}, \bar{a})$, then there is $\bar{b} \subset cl^{\mathcal{M}}(\bar{c})$ such that $\mathcal{M} \vDash \psi(\bar{c}, \bar{b})$.*

Proof. It is obvious that the statement above implies Condition B. The other direction follows by an easy induction argument. More clearly, suppose $\bar{a} = (a_0, \ldots, a_n)$. If $\mathcal{M} \models \psi(\bar{c}, \bar{a})$, then $\mathcal{M} \models \exists \bar{y} \psi(\bar{c}, \bar{y}, a_0)$. By Condition B, we can find $\hat{a}_0 \in cl^{\mathcal{M}}(\bar{c})$ such that $\mathcal{M} \models \exists \bar{y} \psi(\bar{c}, \bar{y}, \hat{a}_0)$. This means that for some tuple $\bar{m}_0$, $\mathcal{M} \models \psi(\bar{c}, \bar{m}_0, \hat{a}_0)$. Note that $|\bar{m}_0| = |\bar{a}| - 1$. Let $\bar{c}_1 = \hat{a}_0 \bar{c}$. Again using condition B, we know that there exists $\hat{a}_1 \in cl^{\mathcal{M}}(\bar{c}_1) (= cl^{\mathcal{M}}(\bar{c}))$ and $\bar{m}_1$ such that $|\bar{m}_1| = |\bar{m}_0| - 1$ and $\mathcal{M} \models \psi(\bar{c}_1, \bar{m}_1, \hat{a}_0, \hat{a}_1)$. Continuing the same way, we end up with $\mathcal{M} \models \psi(\bar{c}_1, \hat{a}_0, \ldots, \hat{a}_n)$ for some $\hat{a}_0, \ldots, \hat{a}_n \in cl^{\mathcal{M}}(\bar{c})$. □

Theorem 4.2.6. *Let K be a class of computable r.i.c.e. pregeometries which is closed under isomorphism. Suppose that K contains an infinite dimensional computable pregeometry with computable basis that satisfies Condition B. Then, the isomorphism problem for the class K is Π_3-hard (i.e. every Π_3 set is m-reducible to $I(K)$).*

Before we prove the theorem, we introduce some notation. Suppose $\theta' = \theta(m_1, \ldots, m_i)$ is a sentence in the diagram of some structure $\mathfrak{M}$, and that f is some function with domain that contains $m_1, \ldots, m_i$. Then, we write $f(\theta')$ to mean $\theta(f(m_1), \ldots, f(m_2))$. If $\overline{m} = (m_1, \ldots, m_i)$ is a tuple of elements in $dom(f)$, then we write $f(\bar{m})$ to mean the tuple $(f(m_1), \ldots, f(m_i))$. Accordingly, $\theta(f(\overline{m}))$ and $f(\theta(\overline{m}))$ mean the same thing. We also adopt similar definitions for $f^{-1}(\theta')$ and $f^{-1}(\overline{m})$.

Proof. Let $\mathcal{M} \in K$ be computable such that $M = \{m_i\}_{i \in \mathbb{N}}$, and let $\{a_i\}_{i \in \mathbb{N}} \subset M$ be an infinite computable basis of $\mathcal{M}$. We uniformly build a computable sequence of structures $\mathcal{N}_k$ such that $\mathcal{N}_k$ is isomorphic to $\mathcal{M}_{|\overline{W_k}|}$, where $\mathcal{M}_t = cl^{\mathcal{M}}(a_0, \ldots, a_t)$ for every natural number t and $\mathcal{M}_\infty = \mathcal{M}$. This implies that $\overline{\mathbf{Cof}}$, which is Π_3-complete, is 1-reducible to $I(K)$.

Fix an arbitrary k. We build a computable structure $\mathcal{N} (= \mathcal{N}_k)$ with universe $N = \{n_i : i \in \mathbb{N}\}$ by defining its atomic diagram stage by stage. To do so, at every stage s, we will define a partial computable function f_s from $\mathcal{M}$ to a subset of N. As s gets bigger, $Im(f_s)$ gets bigger and eventually $\cup_s Im(f_s) = N$. We also make sure that the limit $\lim_s f_s$ exists and is an isomorphism from $\mathcal{M}_{|\overline{W_k}|}$ onto $\mathcal{N}$. The partial functions f_s will be used to define the relations of $\mathcal{N}$ stage by stage.

Let $\{\theta_e\}_{e\in\mathbb{N}}$ be a computable enumeration of the atomic sentences of $\mathcal{N}$. We construct f and D (D will be the atomic diagram of $\mathcal{N}$) by stages to meet the following conditions:

$P_e^1 : \theta_e \in D$ or $\neg\theta_e \in D$,
$P_e^2 : n_e \in Im(f)$,
$Q_e : dom(f) \supseteq cl^{\mathcal{M}}(a_0, \ldots, a_e)$ if and only if $|\overline{W_k}| \geq e$.

We also make sure that, once θ_e (or $\neg\theta_e$) is enumerated in D, it stays in D forever; hence D is computable. At every stage s, f_s is injective and is a partial homorphism.

We say that P_e^1 requires attention at an odd stage $s+1 = 2p+1$ if neither θ_e nor $\neg\theta_e$ is in D_s, and the n_i occurring in θ_e have $i \leq p$. The requirement P_e^1 will not require attention at even stages. We also say that Q_e requires attention at an even stage $s+1 = 2p$ if $z_{e,p-1} \in W_{k,p}$ (where for every natural t, $\overline{W_{k,t}} = \{z_{0,t} < z_{1,t} < z_{2,t} < \ldots\}$). In other words, the requirement Q_e requires attention at an even stage $s+1 = 2p$ if the e^{th} element of the complement of $W_{k,p-1}$ enters $W_{k,p}$. The requirement Q_e will not require attention at odd stages.

We start with $f_0 = \{\langle a_0, n_0\rangle\}$ and $D_0 = \varnothing$. Suppose f_s and D_s are now defined such that $Im(f_s) = \{n_i : 0 \leq i \leq \frac{s}{2}\}$ and every n_i that occurs in a sentence in D_s is such that i is less than $\frac{s}{2}$.

At odd stage $s+1 = 2p+1$, look for the least $e \leq p$ such that P_e^1 requires attention. If such e does not exist, leave $D_{s+1} = D_s$. If such e exists, check if $\mathcal{M} \vDash f_s^{-1}(\theta_e)$. If so, let $D_{s+1} = D_s \cup \{\theta_e\}$. Otherwise, let $D_{s+1} = D_s \cup \{\neg\theta_e\}$.

Set $f_{s+1} = f_s$ (we work on it when $s+1$ is even).

At even stage $s+1 = 2p$, we will work on Q_e for $e > 0$. At the end of the structure, the reader will be able to see that Q_0 is automatically satisfied because $f_s(a_0) = n_0$ for all s. Let $\Theta_s(n_0, \ldots, n_{p-1})$ be the finite conjunction of all the sentences in $D_s \cup \{n_i \neq n_j : i \neq j \text{ and } i, j \leq p-1\}$ (we took that union to ensure later that f_{s+1} is injective when we define it). For ease, let $\overline{n}$ denote $(n_0, \ldots, n_{p-1})$.

Look for the least $e > 0$ such that $e < p$ and Q_e requires attention. If

such e does not exist, define $f_{s+1} = f_s \cup \{\langle m, n_p\rangle\}$ where m is the least that is not in $dom(f_s)$. Otherwise, let us write the tuple $\overline{n}$ as $(\overline{c}, \overline{d})$ where $\overline{c}$ are the components that appear in $f_s(cl_s^{\mathcal{M}}(a_0, \ldots, a_e))$ and $\overline{d}$ are the rest. To make our construction easier to follow, assume that our computable approximation of the operator $cl^{\mathcal{M}}$ is such that, for every tuple $\overline{v} \subset M$, $\overline{v} \subseteq cl_0^{\mathcal{M}}(\overline{v})$, and $cl_s^{\mathcal{M}}(\overline{v}) \subseteq cl_{s+1}^{\mathcal{M}}(\overline{v})$ for every s. Moreover, for all tuples $\overline{v}, \overline{u} \subset M$, we assume (without loss of generality) that our enumeration of $cl^{\mathcal{M}}(\overline{v})$ is fast enough so that, if at any part of the construction before a stage t, we find out that the elements of $\overline{u}$ belong to $cl^{\mathcal{M}}(\overline{v})$, then $\overline{u} \subseteq cl_t^{\mathcal{M}}(\overline{v})$. This last sentence relates to what is going on in the next paragraph when we look ahead (possibly) in the enumeration of $cl^{\mathcal{M}}(\overline{v})$ (speed it up) to apply Condition B (using Lemma 4.2.5).

We now have that $\mathcal{M} \vDash \Theta_s(f_s^{-1}(\overline{c}), f_s^{-1}(\overline{d}))$. Then, by Condition B, there exists $\overline{w} \subset cl^{\mathcal{M}}(f_s^{-1}(\overline{c})) \subseteq cl^{\mathcal{M}}(a_0, \ldots, a_e)$ such that $\mathcal{M} \vDash \Theta_s(f_s^{-1}(\overline{c}), \overline{w})$. Assume that $\overline{w}$ is the first such tuple to get enumerated into $cl^{\mathcal{M}}(f_s^{-1}(\overline{c}))$. Note that according to the previous paragraph, we assume that $\overline{w} \subseteq cl_s^{\mathcal{M}}(f_s^{-1}(\overline{c}))$.

Set $f_{s+1}(\overline{w}) = \overline{d}$ and $f_{s+1}(\overline{c}) = f_s(\overline{c})$. Define $f_{s+1}(m) = n_p$ where m is the least that is not yet in $dom(f_{s+1})$.

Set $D_{s+1} = D_s$. This completes the construction.

We start our verification by observing the following:

(1) Once an element from $\mathcal{N}$ becomes in $Im(f_s)$, at some stage s, it will remain in $Im(f_t)$ for all $t > s$.

(2) For every even stage $s + 1 = 2p$ and every e, if $z_{e,p-1}$ gets enumerated into W_k (i.e. Q_e requires attention at stage $s + 1$), we set $dom(f_{s+1})$ to be the union of a subset of $cl^{\mathcal{M}}(a_0, \ldots, a_e)$ with a singleton set. This is because we made $f_{s+1}^{-1}(\{n_0, \ldots, n_{p-1}\}) \subseteq cl_s^{\mathcal{M}}(a_0, \ldots, a_{e-1})$, and we made $f_{s+1}^{-1}(n_p)$ equal to the next fresh element in M.

(3) Suppose s is a stage after which $Q_{e'}$ does not require attention for $e' < e$ (for some $e \geq 1$). For $n_i \in Im(f_s)$ (i.e. for $i \leq \frac{s}{2}$), once $f_{s+1}^{-1}(n_i)$ is defined to be an element from $cl^{\mathcal{M}}(a_0, \ldots, a_e)$, we will have that $f_t^{-1}(n_i) = f_{s+1}^{-1}(n_i)$ for all $t > s$. For, we argue by induction. Assume for

some $t > s$ that $f_{t'}^{-1}(n_i) = f_{s+1}^{-1}(n_i)$ for all t' such that $s < t' \leq t$. Assume towards a contradiction that $f_{t+1}^{-1}(n_i) \neq f_t^{-1}(n_i)$. This means that, at even stage $t+1$, $Q_{e'}$ requires attention for some e', which has to be $\geq e$, and $f_t^{-1}(n_i) \notin cl_t^{\mathcal{M}}(a_0, \ldots, a_{e'})$. But, this will not happen because of our assumption regarding the speed of enumerating $cl^{\mathcal{M}}$. Indeed, our enumeration of $cl^{\mathcal{M}}$ guarantees that $f_t^{-1}(n_i) = f_{s+1}^{-1}(n_i) \in cl_{s+1}^{\mathcal{M}}(a_0, \ldots, a_e) \subseteq cl_t^{\mathcal{M}}(a_0, \ldots, a_{e'})$.

From the list of observations above, we can prove the following essential claims to complete our proof.

Claim 1: For every $i \in \omega$, once $n_i \in Im(f_s)$ for some stage s, we have that $n_i \in Im(f_t)$ for all $t > s$. Moreover, we have that $\lim_s f_s^{-1}(n_i)$ exists which implies that there is a function f such that $f = \lim_s f_s$ from $\cup_{s \in \omega} f_s^{-1}(N)$ onto $\mathcal{N}$.

Proof of Claim 1: It easy to see that the first part of the claim is true. For the second part, we will consider two cases:

Case 1: $|\overline{W_k}| = e < \infty$. In this case, $\lim_s z_{e',s}$ exists for all (and only for) $e' < e$. This means that there is a stage s such that, in the stages $t > s$, none of $z_{e',t}$ (for $e' < e$) will get enumerated into W_k. In other words, for $e' < e$, $Q_{e'}$ will not require attention after stage s. But, in the same time, at infinitely many stages t, $z_{e,t}$ will get enumerated into W_k. Let $t_0 > s$ be such a stage, and let $\overline{n} = \{n_i : i \leq \frac{t_0}{2}\}$. Then, from observation 2 above, f_{t_0+1} will be defined so that $f_{t_0+1}^{-1}(\{n_0, \ldots, n_{t_0}\})$ is a subset of $cl^{\mathcal{M}}(a_0, \ldots, a_e)$. From observation 3, we will have that $f_t^{-1}(\overline{n}) = f_{t_0+1}^{-1}(\overline{n})$ for all $t > t_0$. Now, since t_0 can be arbitrarily large, we have that $\lim_s f_s^{-1}(n)$ exists for every $n \in N$. Moreover, that limit belongs to $cl^{\mathcal{M}}(a_0, \ldots, a_e)$.

Case 2: $|\overline{W_k}| = \infty$. In this case, for every e, $\lim_s z_{e,s}$ exists. This means that, for every e, there is a stage $s(e)$ (for which $z_{e,t} = z_{e,s(e)}$ for all $t > s(e)$) which is the last stage at which Q_e requires attention, and after which $Q_{e'}$ does not require attention for all $e' \leq e$. It follows from observation 3, through the same reasoning in case 1, that $\lim_s f_s^{-1}(n_i)$ exists for $i \leq n_{s(e)}$ for every e. Since $s(e) \to \infty$ as $e \to \infty$ (obvious), it follows that $\lim_s f_s^{-1}(n_i)$ exists for

every $i \in \mathbb{N}$.

Claim 2: $dom(f) = \mathcal{M}_{|\overline{W_k}|}$.

Proof of Claim 2: We consider our two cases:

Case 1: $|\overline{W_k}| = e < \infty$. As explained in the previous note, in infinitely many stages t, $z_{e,t}$ will get enumerated into $W_{|\overline{W_k}|}$. This implies that, for every $i \in \omega$, there is a stage s such that $f_s^{-1}(n_i) \in \mathcal{M}_{|\overline{W_k}|}$. We have that $f_t^{-1}(n_i) \in \mathcal{M}_{|\overline{W_k}|}$ for all $t > s$ (see observations 2 and 3). Hence, $f^{-1}(n_i) \in \mathcal{M}_{|\overline{W_k}|}$. Conversely, every element in $\mathcal{M}_{|\overline{W_k}|}$ is in $dom(f)$. Indeed, at every stage $2s + 1$ in which Q_e requires attention (and $Q_{e'}$ no longer requires attention for $e' < e$), the members of the first available tuple in the enumeration of $cl^{\mathcal{M}}(a_0, \ldots, a_e) = \mathcal{M}_{|\overline{W_k}|}$ get enumerated into $dom(f_{s+1})$. We end up considering all tuples because every element of $\mathcal{M}_{|\overline{W_k}|}$ appears within a sentence in the atomic diagram of $\mathcal{M}_{|\overline{W_k}|}$. Since Q_e requires attention infinitely often, we eventually have that $dom(f) = \mathcal{M}_{|\overline{W_k}|}$.

Case 2: $|\overline{W_k}| = \infty$. In this case, it is clear that $dom(f) \subseteq \mathcal{M}_{|\overline{W_k}|} = \mathcal{M}$. Conversely, every element in $\mathcal{M}$ gets enumerated into $dom(f)$ eventually. For, it is enough to show that, for every e, every element of $cl^{M}(a_0, \ldots, a_e)$ gets enumerated into $Im(f_s)$, for some s, and remains in $dom(f_t)$ for all $t > s$. Indeed, let s be a stage after which $Q_{e'}$ stops receiving attention for all $e' < e$. Then, for every $t > s$, if $m \in cl^{\mathcal{M}}(a_0, \ldots, a_e)$ gets enumerated into $dom(f_t)$, then $m \in dom(f_{t'})$ for all $t' > t$ (i.e. remains in the image forever as explained in observation 3). Since we always enumerate the first available fresh element from $\mathcal{M}$ or from $cl^{\mathcal{M}}(a_0, \ldots, a_{e'})$ for some $e' \geq e$, and since every element of $\mathcal{M}$ appears within a sentence in the atomic diagram of $\mathcal{M}$, we have that every element which is not enumerated in $dom(f_s)$ yet will be enumerated in $dom(f_t)$ for some $t > s$ and will remain in $dom(f_{t'})$ for all $t' > t$.

Claim 3: The set $D = \cup_s D_s$ is computable.

Proof of Claim 3: Indeed, for every $l \in \omega$, there is some stage $s+1$ at which either θ_l or $\neg\theta_l$ enters D_{s+1}. Note that this happens based on whether $\mathcal{M} \vDash f_s^{-1}(\theta_l)$ or not. In the construction, we managed to maintain that satisfiability

at later stages by applying Condition B. To be clearer, we have that $\mathcal{M} \models f_s^{-1}(\theta_l)$ implies that $\mathcal{M} \models f_t^{-1}(\theta_l)$ for $t \geq s$ (and the same for $\neg\theta_l$). This implies that it will never happen that both θ_l and $\neg\theta_l$ end up in D. So eventually, every atomic sentence or its negation will get enumerated in D; hence D is computable.

Finally, it is not hard to see that f is an isomorphism. As mentioned in the proof of Claim 3, at every stage $s+1$, f_{s+1} is chosen to preserve the sentences included in D_s (behavior of a homomorphism). Also we chose f_{s+1} to explicitly preserve $n \neq m$ for all different $n, m \in Im(f_s)$. Accordingly, f_t has to be injective all stages t. From this and from Claims 1 and 2, it is obvious that f is an isomorphism from $\mathcal{M}_{|\overline{W_k}|}$ onto $\mathcal{N}$. This completes our verification.

□

Chapter 5

Separating the UTO and LTO

Recall from the introduction that Jockusch and Soare introduced the notion of Turing ordinal of a class of computable structures as a measurement of the difficulty of coding information into the structures of a given class.

The following are examples of known Turing ordinals for some classes:

(1) Abelian groups, graphs, partial orderings and lattices have $TO = 0$ (see [35]).

(2) Equivalence Structures have $TO = 1$ (see [35],[31]).

(3) Linear ordering $TO = 2$ (see [35],[24]).

(4) Boolean algebras $TO = \omega$ (see [7]).

In [25], Knoll used the linear orderings defined by Ash, Jockusch and Knight in [3] to give examples of classes with Turing ordinal α for every computable ordinal α.

Recall the definitions of the Upper and Lower Turing Ordinals from the introduction.

Definition 5.0.1. Let K be a class of first-order structures. We call a computable ordinal β the *upper Turing ordinal (abbreviated as UTO)* of K if it is the least ordinal such that the first condition of the *TO* definition holds. More precisely:

(1) every degree $\geq \mathbf{0}^{(\beta)}$ is the β^{th} jump degree of a structure from K, and

(2) for all $\eta < \beta$, there exists a degree $\mathbf{d} \geq \mathbf{0}^{(\eta)}$ which is not the η^{th} jump degree of a structure from K.

Definition 5.0.2. Let K be a class of first-order structures. We call a computable ordinal α the *lower Turing ordinal (abbreviated as LTO)* of K if it is the greatest ordinal such that the second condition of the *TO* definition holds. More precisely:

(1) there is a degree $\mathbf{d} > \mathbf{0}^{(\alpha)}$ such that $\mathbf{d}$ is the α^{th} jump degree of a structure from K, and

(2) for all $\eta < \alpha$, the only possible η^{th} jump degree of a structure from K is $\mathbf{0}^{(\eta)}$.

We will show that for all computabe ordinals α, β such that $\alpha < \beta$, there is a class of structures with *UTO* = β and *LTO* = α. We will use combinations of the classes given by Knoll in [25], graphs, and equivalence structures.

It is worth mentioning that the classes we construct here are "natural" in the sense that they are not too hard to axiomatize. Axiomatizability is not our concern here but one can show that the way we derive our classes from those defined by Knoll maintains the Borelness (definability by an $\mathcal{L}_{\omega_1,\omega}$ sentence).

The following are some definitions and notation to make describing our classes easier.

Definition 5.0.3. We classify the computable ordinals ≥ 2 into three types:

(1) Successor ordinal of type I: which is either a finite even number ≥ 2 or $\beta+$ odd (for some limit ordinal β).

(2) Successor ordinal of type II: which is either a finite odd number ≥ 3 or $\beta+$ even (for some limit ordinal β).

(3) Limit ordinal.

For every computable ordinal $\alpha \geq 2$ and every set S of natural numbers, there is a linear order $\mathcal{A}_\alpha(S)$ (using the notation in [25]) which is defined based on the type of α according to Definition 5.0.3. We will not need here to fully describe the linear orderings, but a full description can be found in Chapter 6 of [25].

Definition 5.0.4. [23] Let α be a computable ordinal. A set $S \subseteq \omega$ is *α-generic* if for each Σ_α set $X \subseteq 2^{<\omega}$, there is some finite $\sigma \subset S$ such that either $\sigma \in X$ or else there is no finite $\tau \supseteq \sigma$ such that $\tau \in X$.

Recall that for every computable limit ordinal γ, there is a fundamental sequence of computable ordinals $(\gamma_k)_{k\in\omega}$ that converges to γ. We introduce here the following defintion:

Definition 5.0.5. Let γ be a computable limit ordinal with fundamental sequence $(\gamma_k)_{k\in\omega}$. The *adjusted* fundamental sequence of γ, denoted $(\gamma'_k)_{k\in\omega}$ is defined as follows:

(1) If γ_k is finite, then $\gamma'_k = \min\{i : i \text{ is even}, i \geq 4, i \geq \gamma_k, i \geq \gamma'_{k-1}\}$.

(2) If γ_k is infinite, then let β be the greatest limit ordinal $\leq \gamma_k$ and define $\gamma'_k = \min\{\eta : \eta = \beta + i, i \text{ is odd }, \eta \geq \gamma_k, \eta \geq \gamma'_{k-1}\}$.

Definition 5.0.6. Let S be a set of natural numbers and let $S_n = \{k : \langle n, k\rangle \in S\}$ for every natural number n. Let $\gamma \geq 2$ be a computable ordinal. We define $P_\gamma(S)$ to be the set theoretic predicate of S that stands for:

(1) "$S = S$" if γ is a successor ordinal of type I,

(2) "S is γ-generic" if γ is a successor ordinal of type II,

(3) "$S_k \not\leq_T (S_0 \oplus \ldots \oplus S_{k-1})^{(\gamma_k)}$ for all $k \in \omega$" if γ is a limit ordinal.

We have the following fundamental results (for proofs, see [25]).

Theorem 5.0.7 (Ash, Jockusch and Knight)**.** *For every $S \subseteq \omega$, and every computable ordinal $\gamma \geq 2$, we have that* $\mathbf{Spec}(\mathcal{A}_\gamma(S)) =:$

(1) $\{\mathbf{deg}(D) : \mathbf{deg}(S) \leq_T D^{(\gamma)}\}$ *if γ is a successor ordinal of type I;*

(2) $\{\mathbf{deg}(D) : S$ *is c.e. in* $D^{(\gamma-1)}\}$ *if γ is a successor ordinal of type II; and*

(3) $\{\mathbf{deg}(D) : S_n \leq_T D^{(\gamma'_n)}$ *uniformly in* $n\}$ *if γ is a limit ordinal and γ'_n is an adjusted fundamental sequence with limit γ.*

We note an immediate consequence of Theorem 5.0.7.

Corollary 5.0.8. *For every computable ordinal $\gamma \geq 2$,*

$$\mathbf{Spec}(\mathcal{A}_\gamma(S)) \subseteq \{\mathbf{deg}(D) : S \leq_T D^{(\gamma)}\}.$$

Theorem 5.0.9 (Ash, Jockusch and Knight)**.** *For every computable ordinal $\gamma \geq 2$, the class $C_\gamma = \{\mathcal{A}_\gamma(S) : S \subseteq \omega \wedge P_\gamma(S)\}$ has Turing ordinal γ.*

Remark 5.0.10. For every computable ordinal $\gamma \geq 2$, the proof of Theorem 5.0.9 (see [25]) explains how, for a given degree $\mathbf{d}$, we choose the set S to realize $\mathbf{d}$ as a degree of the structure $\mathcal{A}_\gamma(S)$. We mention here the connection between the degree of $\mathcal{A}_\gamma(S)$ and $\mathbf{deg}(S)$ as it will be useful later.

(1) If γ is a successor ordinal of type I and S is any set of natural numbers such that $\mathbf{deg}(S) \geq \mathbf{0}^{(\gamma)}$, we have that $\mathbf{deg}(S)$ is the γ^{th} jump degree of $\mathcal{A}_\gamma(S)$ (for proof, see Theorem 6.1.9 in [25]).

(2) If γ is a successor ordinal of type II or is a limit ordinal, and S is γ-generic, then the γ^{th} jump degree of $\mathcal{A}_\gamma(S)$ is $\mathbf{deg}(S)^{(\gamma)}$ (for proof, see Theorems 6.1.13, 6.1.20 in [25]).

Now let us define some building blocks which we will use to build our classes. In the statement of Theorem 5.0.9, we defined the notation C_γ for $\gamma \geq 2$. It will be helpful to denote the class of equivalence structures by C_1.

For every computable ordinal $\gamma \geq 2$, and every computable ordinal β, let $C(\gamma, \beta)$ be the class $\{\mathcal{A}_\gamma(S) : S \subseteq \omega \wedge P_\gamma(S) \wedge S \not\leq_T \mathbf{0}^{(\beta)}\}$.

We know that for any set X of natural numbers, there is a graph G_X that has degree $\mathbf{deg}(X)$. Let $C(0, \beta)$ be the class $\{G_X : X \subseteq \omega \wedge X \not\leq_T \mathbf{0}^{(\beta)}\}$.

Let $\mathcal{U}_\beta$ be an equivalence structure on an infinite domain (i.e. (ω, E) for some equivalence relation E on ω) such that $\mathcal{U}_\beta$ has first jump degree $\mathbf{0}^{(\beta+1)}$. Such a structure exists because of the fact that the Turing ordinal for the class of equivalence structures is 1. For every computable ordinal β, let $C(1, \beta)$ be the class $\{\mathcal{B} : \mathcal{B} \cong \mathcal{U}_\beta\}$.

Let $C_\alpha^\beta = C_\beta \cup C(\alpha, \beta)$. We will show that C_α^β is the class we are looking for.

Lemma 5.0.11. *Let α, β and η be computable ordinals such that $\eta \geq \alpha$. Then, any degree $\leq \mathbf{0}^{(\beta)}$ cannot be the η^{th} jump degree of any structure from $C(\alpha, \beta)$.*

Proof. When $\alpha < 2$, the lemma is clear. Suppose $\alpha \geq 2$. Let D be a subset of natural numbers such that $\mathbf{deg}(D)$ is in the degree spectrum of some structure from $C(\alpha, \beta)$. Then, for some $S \subseteq \omega$ such that $S \not\leq_T \mathbf{0}^{(\beta)}$, $\mathbf{deg}(D)$ belongs to $\mathbf{Spec}(\mathcal{A}_\alpha(S))$. We show that, for every computable ordinal $\eta \geq \alpha$, $\mathbf{deg}(D)^{(\eta)} \not\leq \mathbf{0}^{(\beta)}$.

Indeed, since $S \not\leq_T \mathbf{0}^{(\beta)}$, by Corollary 5.0.8 we have that $D^{(\alpha)} \not\leq_T \mathbf{0}^{(\beta)}$. This implies that $D^{(\eta)} \not\leq_T \mathbf{0}^{(\beta)}$ for any computable ordinal $\eta \geq \alpha$. □

The following readily follows from the last lemma.

Corollary 5.0.12. *Let α, β and η be computable ordinals such that $\alpha \leq \eta < \beta$. There exists at least one degree which is strictly greater than $\mathbf{0}^{(\eta)}$ and cannot be realized as the η^{th} jump degree of a structure in $C(\alpha, \beta)$.*

Proof. An example of such a degree is $\mathbf{0}^{(\beta)}$. By Lemma 5.0.11, $\mathbf{0}^{(\beta)}$ cannot be the η^{th} jump degree of any structure from $C(\gamma, \beta)$. □

Now let us prove the following:

Proposition 5.0.13. *Let α, β be computable ordinals such that $\alpha < \beta$. The class C_α^β has UTO = β and LTO = α.*

Proof. Let us start by showing that *UTO* = β. It is easy to see that *UTO* $\leq \beta$. This is because if $\mathbf{d} \geq \mathbf{0}^{(\beta)}$, then $\mathbf{d}$ can be realized as the β^{th} jump degree of a structure in C_β (a subclass of C_α^β) which has Turing ordinal β by Theorem 5.0.9.

It remains to show that *UTO* $\geq \beta$. This can be done by showing that, for every computable ordinal $\eta < \beta$, there exists at least one degree $\geq \mathbf{0}^{(\eta)}$ which cannot be realized as the η^{th} jump degree of a structure in C_α^β. This is clearly equivalent to showing that there exists at least one degree which is strictly greater than $\mathbf{0}^{(\eta)}$ and cannot be realized as the η^{th} jump degree of a structure in $C(\alpha, \beta)$. The latter follows immediately from Corollary 5.0.12.

Let us now prove that *LTO* = α. To see that *LTO* $\geq \alpha$, first note that for every computable ordinal $\eta < \alpha$, $\mathbf{0}^{(\eta)}$ is realizable as the η^{th} jump degree of a structure in C_β. Moreover, these are the only degrees realizable as the η^{th} jump degree of a structure in C_β. Also it is clear that for $\eta < \alpha$, if $\mathbf{d} > \mathbf{0}^{(\eta)}$,

then $\mathbf{d}$ is not relaizable as the η^{th} jump degree of any structure in $C(\alpha, \beta)$. This is because $C(\alpha, \beta)$ is a subclass of C_α which has Turing ordinal = α.

To see that *LTO* $\leq \alpha$, we need to realize a degree $> \mathbf{0}^{(\alpha)}$ as the α^{th} jump degree of some structure in C_α^β. When $\alpha = 0$ or 1, consider for example the degree $\mathbf{0}^{(\beta+1)}$. That degree is obviously $> \mathbf{0}^{(\alpha)}$ and realizable as the α^{th} jump degree of a structure from $C(\alpha, \beta) \subseteq C_\alpha^\beta$.

When $\alpha \geq 2$, we need to consider the different cases of α.

Case 1: α is the successor ordinal of type I. In this case, by Remark 5.0.10, $\mathcal{A}_\alpha(S)$ has α^{th} jump degree $\mathbf{deg}(S)$. We can choose S to have any degree $\not\leq \mathbf{0}^{(\beta)}$. This will guarantee that $\mathcal{A}_\alpha(S)$ belongs to $C(\alpha, \beta) \subseteq C_\alpha^\beta$ and that it has a non-trivial α^{th} jump degree (different from $\mathbf{0}^{(\alpha)}$).

Case 2: α is the successor ordinal of type II or is a limit ordinal. Again, from Remark 5.0.10, for an α-generic set S, $\mathcal{A}_\alpha(S)$ has α^{th} jump degree $\mathbf{deg}(S^{(\alpha)})$. Again, we can choose S to have any degree $\not\leq \mathbf{0}^{(\beta)}$ which implies that we have a structure in $C(\alpha, \beta) \subseteq C_\alpha^\beta$ with non-trivial α^{th} jump degree.

□

Bibliography

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

[2] Ash, C., Knight, J., Manasse, M., and Slaman, T. (1989). Generic copies of countable structures. *Ann. Pure Appl. Logic*, 42(3):195–205.

[3] Ash, C. J., C. G. Jockusch, J., and Knight, J. F. (1990). Jumps of orderings. *Trans. Amer. Math. Soc.*, 319(2):573–599.

[4] Ash, C. J. and Knight, J. (2000). *Computable structures and the hyperarithmetical hierarchy*, volume 144 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam.

[5] Bazhenov, N. and Marchuk, M. (2018). Degrees of categoricity for prime and homogeneous models. In *Sailing routes in the world of computation*, volume 10936 of *Lecture Notes in Comput. Sci.*, pages 40–49. Springer, Cham.

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

[7] C. G. Jockusch, J. and Soare, R. I. (1994). Boolean algebras, Stone spaces, and the iterated Turing jump. *J. Symbolic Logic*, 59(4):1121–1138.

[8] Calvert, W. (2004). The isomorphism problem for classes of computable fields. *Arch. Math. Logic*, 43(3):327–336.

[9] Calvert, W. (2005). The isomorphism problem for computable abelian p-groups of bounded length. *J. Symbolic Logic*, 70(1):331–345.

[10] Calvert, W. and Knight, J. F. (2006). Classification from a computable viewpoint. *Bull. Symbolic Logic*, 12(2):191–218.

[11] Chisholm, J. (1990). Effective model theory vs. recursive model theory. *J. Symbolic Logic*, 55(3):1168–1191.

[12] Csima, B. F., Deveau, M., Harrison-Trainor, M., and Mahmoud, M. A. (2019). Degrees of categoricity above limit ordinals. *Submitted*.

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

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

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

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

[17] Goncharov, S. S. and Naĭt, D. (2002). Computable structure and antistructure theorems. *Algebra Logika*, 41(6):639–681, 757.

[18] Groszek, M. J. and Slaman, T. A. (2007). Moduli of computation. Talk presented at the Conference on Logic, Computability and Randomness, Buenos Aires, Argentina.

[19] Harrison-Trainor, M. (2018). Computable valued fields. *Arch. Math. Logic*, 57(5-6):473–495.

[20] Harrison-Trainor, M., Melnikov, A., and Montalbán, A. (2015). Independence in computable algebra. *J. Algebra*, 443:441–468.

[21] Hirschfeldt, D. R., Khoussainov, B., Shore, R. A., and Slinko, A. M. (2002). Degree spectra and computable dimensions in algebraic structures. *Ann. Pure Appl. Logic*, 115(1-3):71–113.

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

[23] Johnson, J., Knight, J. F., Ocasio, V., and VanDenDriessche, S. (2013). An example related to Gregory's theorem. *Arch. Math. Logic*, 52(3-4):419–434.

[24] Knight, J. F. (1986). Degrees coded in jumps of orderings. *J. Symbolic Logic*, 51(4):1034–1042.

[25] Knoll, C. (2013). *Complexity of Classes of Structures*. PhD thesis.

[26] Kuske, D., Liu, J., and Lohrey, M. (2013). The isomorphism problem on classes of automatic structures with transitive relations. *Trans. Amer. Math. Soc.*, 365(10):5103–5151.

[27] Lempp, S., McCoy, C., Miller, R., and Solomon, R. (2005). Computable categoricity of trees of finite height. *J. Symbolic Logic*, 70(1):151–215.

[28] Mahmoud, M. A. (2019a). Degrees of categoricity of trees and the isomorphism problem. *To appear in the J. MLQ*.

[29] Mahmoud, M. A. (2019b). The isomorphism problem for pregeometries. *In preparation*.

[30] Marker, D. (2006). *Model theory: an introduction*, volume 217. Springer Science & Business Media.

[31] Montalbán, A. (2012). Counting the back-and-forth types. *J. Logic Comput.*, 22(4):857–876.

[32] Montalbán, A. (2018). Coding and definability in computable structures. *Notre Dame J. Form. Log.*, 59(3):285–306.

[33] Morozov, A. S. (1993). Functional trees and automorphisms of models. *Algebra i Logika*, 32(1):54–72, 112.

[34] Nies, A. (1996). Undecidable fragments of elementary theories. *Algebra Universalis*, 35(1):8–33.

[35] Richter, L. J. (1981). Degrees of structures. *J. Symbolic Logic*, 46(4):723–731.

[36] Soare, R. I. (2016). *Turing computability*. Theory and Applications of Computability. Springer-Verlag, Berlin.

[37] Tent, K. and Ziegler, M. (2012). *A course in model theory*, volume 40 of *Lecture Notes in Logic*. Association for Symbolic Logic, La Jolla, CA; Cambridge University Press, Cambridge.