CHARACTERIZATIONS FOR COMPUTABLE STRUCTURES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Walker McMillan White

August 2000

CHARACTERIZATIONS FOR
COMPUTABLE STRUCTURES

Walker McMillan White, Ph.D.
Cornell University 2000

A major theme in computable model theory is the study of necessary and sufficient conditions for the existence of certain types of computable structures. These characterizations may either be syntactic, as in the work of Millar [36], or semantic, as in the work of Goncharov [16]. Presented in this dissertation is a general framework for proving syntactic characterizations for the existence of computable models of various theories. This framework is used to show that an axiomatizable $\forall_2$ theory has a computable, existentially closed model if and only if it has an existentially conservative 1-completion for which the set of universal theorems is decidable. Several other uses of this framework are given; one answers an open question of Baldwin and Kueker [4] in the classical model theory of existentially closed, algebraically prime models. Semantic characterizations of computable structures are also investigated. By analyzing the computational complexity of various classes of computable structures, it is shown that various model theoretic properties have no essentially simpler characterization. For example, it is shown that the classes of computable homogeneous structures, computable atomic structures, and computable computably saturated structures are all $\Pi^0_{\omega+2}$-complete. Other results include the fact that the class of hyperarithmetically categorical structures is Π^1_1-complete, and that the set of computably categorical structures is Π^0_4-hard.

Biographical Sketch

Walker McMillan White was born in Winston-Salem, North Carolina. He later moved to Oak Ridge, Tennessee, where he attended Oak Ridge High School, graduating in 1989. He received a B.A. in mathematics from Dartmouth College in 1993 and an M.S. in computer science from Cornell University in 1998. In August 2000, he will start a tenure-track position teaching mathematics and computer science at the University of Dallas in Dallas, Texas.

In memory of my father.

Acknowledgements

It is a daunting task to acknowledge all those who have helped make this dissertation possible. Education is a complex process, and it is not always clear who is the teacher and who is the student. This makes it difficult to identify everyone who has helped me throughout the years.

More than anyone else, I have owe a huge debt of gratitude to my family. My parents were my most influential teachers, and they first taught me the difference between knowledge and information. My passion for learning is a direct result of being raised in an environment where education was always considered a top priority. I thank them all, and I am saddened that my father is not able to see the result of his labors. It is for this reason that I dedicate this dissertation to him, even though all of my family are equally responsible for where I am today.

I have been extremely fortunate to have Richard Shore as an advisor. Both a first-class mathematician and a master teacher, he has been amazingly generous with his time, knowledge, and many resources. I came to Cornell as a student of mathematics; because of Richard, I leave Cornell as a mathematician.

My other advisors at Cornell have been the wonderful graduate students in logic. I truly believe that I have been at Cornell during a "Golden Age" of computability theory. My understanding of logic and mathematics has been greatly shaped and inspired by the many friendships I have established during my time here. While I hesitate to single anyone out, I would most like to thank Denis Hirschfeldt, who has been a valuable friend and collaborator. Without him, Chapter 6 of this dissertation would never exist.

I also thank the other members of my committee, Anil Nerode and Dexter Kozen, for their help and advice.

I thank Carolyn Gordon at Dartmouth and Mike Hilliard at the Oak Ridge National Labs for directing me towards Cornell. Coming to Ithaca has been one of the best decisions that I have ever made in my life.

Finally, I gratefully acknowledge the National Science Foundation for their support throughout my graduate career. They have supported me both independently through the Graduate Research Fellowship program from 1994 to 1998, and through a research assistantship with Richard Shore in 2000.

As I said, it is a difficult task to acknowledge everyone, so I end with thanking everyone who has helped me reach this point. You all know who you are; any omissions are simply a result of my last minute rush to finish this dissertation.

Table of Contents

List of Figures

Chapter 1

Introduction

Over the past few decades, there has been an increasing amount of research in determining when various mathematical constructions can be carried out effectively. One field in which this has been studied extensively is model theory. The articles by Ershov and Goncharov [11] and Harizanov [19] are excellent surveys of the work that has been done in this area since its start in the early 1960's.

In studying the effective content of mathematics, we are particularly interested in *characterizations*. A characterization is an equivalence of two seemingly different mathematical properties. They are useful because often one formulation of a property may be inherently more effective than another.

As a simple example, consider the $n \times n$ matrices over the rationals, $\mathbb{Q}$. If we wanted to determine, using only the definition, whether a matrix A is invertible, we would have to search through all possible matrices until we found its inverse. This is not an effective procedure, as it does not halt when A is not invertible. However, we do know that the invertible matrices are exactly those with nonzero determinant. The determinant of a matrix is computable, and so this fact is a useful characterization of the invertible matrices.

In computable model theory, we are interested in two kinds of characterizations. The first kind is simply an effective version of a standard problem in model theory. In model theory we want to characterize those theories with a particular type of model. In computable model theory we want to characterize those theories for which this particular model can be constructed effectively. This question has been well studied for decidable models, as seen in the survey by Harizanov [19]. Examples include the results of Harrington [20] and Goncharov and Nurtazin [17] for decidable elementarily prime models and Millar [36] for decidable saturated structures.

The other kind of characterization is the one from our example with invertible matrices. We want to know what model theoretic properties are equivalent over computable structures. Examples of such characterizations include the results of Goncharov [16] for computably categorical structures, and the results of Ash and Nerode [3] for intrinsically c.e. relations.

In this work, we will investigate both of these kinds of characterizations. Doing so will require several concepts from both model theory and computability theory. Therefore, in the next section, we will introduce the basic terminology necessary for the later chapters. We will conclude this chapter with a summary of the results of this dissertation.

1.1 Basic notation and terminology

We refer to Soare [46] and Sacks [43] for most of the basic concepts of computability theory. However, even these two works do not agree on all notation, so we need to clarify some of our terminology. ! For each $e \in \omega$ and $X \subseteq \omega$, we let $\{e\}^X$ represent the e^{th} partial computable function with oracle X. If this function is defined on the element $x \in \omega$, we write $\{e\}^X(x) \downarrow$ and let $\{e\}^X(x)$ represent its value. For each $s \in \omega$, we let $\{e\}^X_s$ be the evaluation $\{e\}e^X$ at stage s.

We let $\langle \cdot, \cdot \rangle$ be a one-one function mapping $\omega \times \omega$ onto ω. We will write $\langle a, b, c \rangle$ for $\langle a, \langle b, c \rangle \rangle$, and similarly for longer sequences of natural numbers. Because of this function, we will be very free throughout this dissertation in identifying ω with ω^n for $n \in \omega$, and with $\omega^{<\omega}$, the set of all finite sequences in ω. For each $\sigma = \langle x_0, \dots, x_{k-1} \rangle \in \omega^{<\omega}$, we let $\sigma(i)$ denote x_i and let $|\sigma| = k$ be the length of this sequence. For σ, $\tau \in \omega^{<\omega}$, we let $\sigma\widehat{\ }\tau$ be the concatenation of these two sequences. In other words, if $\sigma = \langle x_0, \dots, x_{k-1} \rangle$ and $\tau = \langle y_0, \dots, y_{\ell-1} \rangle$, then

$$\sigma\widehat{\ }\tau = \langle x_0, \dots, x_{k-1}, y_0, \dots, y_{\ell-1} \rangle \tag{1.1}$$

We define the arithmetic and analytic hierarchy for predicates $\mathcal{R}(f, n)$ on $\omega^\omega \times \omega$ as in Sacks [43]. Similarly, we define the computable ordinals and the hyperarithmetic H-sets, H_a. However, in order to be in better agreement with model theory, we will define the hyperarithmetic hierarchy differently.

Definition 1.1.1. A predicate $\mathcal{R}(f, n)$ is $\Sigma^0_0 = \Pi^0_0$ if it is computable. For any computable ordinal α, $\mathcal{R}(f, n)$ is Σ^0_α if there is a sequence of $\Pi^0_{\beta_k}$ predicates $\mathcal{Q}_k(f, n, x)$, $\beta_k < \alpha$, such that

$$\mathcal{R}(f, n) \quad \Leftrightarrow \quad \bigvee_{k \in \omega} \exists x\, \mathcal{Q}_k(f, n, x) \tag{1.2}$$

Similarly, $\mathcal{R}(f, n)$ is Π^0_α if there is some Σ^0_α predicate $\mathcal{Q}(f, n)$ such that

$$\mathcal{R}(f, n) \quad \Leftrightarrow \quad \neg\mathcal{Q}(f, n) \tag{1.3}$$

It is clear from this definition that, when $\alpha = \beta + 1$ is a successor ordinal, any Σ^0_α predicate $\mathcal{R}(f, n)$ can be written as $\exists \overline{x}\, \mathcal{Q}(f, n, x)$ for some Π^0_β predicate $\mathcal{Q}$; an analogous result holds for Π^0_α predicates. We should also point out that Definition 1.1.1 implies that for any computable ordinal $\alpha \geq \omega$, and any notation $a \in O$ for α, $\Sigma^{H_a}_1 = \Sigma^0_\alpha$. This differs from the definition in both Soare [46] and Rogers [42], where $\Sigma^{H_a}_1 = \Sigma^0_{\alpha+1}$. The hierarchy in Definition 1.1.1 is choosen in order to agree with thes classification of infinitary formulae in Ash and Knight [2].

For the basic notions of model theory, we refer the reader to Hodges [26]. When we speak of a language $\mathcal{L}$, we will assume that it is first order unless otherwise noted. We define the syntactic complexity of first order formula as follows.

Definition 1.1.2. We let $\forall_0 = \exists_0$ be the collection of all quantifier free formulae of $\mathcal{L}$. For $n \geq 1$, we say that $\varphi \in \mathcal{L}$ is an $\exists_n$ *formula* if it is a formula of the form $\exists \overline{x}\, \psi(\overline{x})$, where ψ is an $\forall_{n-1}$ formula. We say that φ is an $\forall_n$ *formula* if $\varphi = \forall \overline{x}\, \psi(\overline{x})$, where ψ is an $\exists_{n-1}$ formula.

We will use this syntactic complexity to analyze both theories, which are deductively closed sets of sentences, and structures. For a theory T and $n \in \omega$, we let $T_{\exists_n}$ be the collection of all $\exists_n$ elements of T. Similarly, we define $T_{\forall_n}$. For any structure $\mathfrak{A}$, the *atomic diagram* is the collection of quantifier free sentences true in $(\mathfrak{A}, a)_{a \in \mathfrak{A}}$. In other words,

$$\operatorname{diag}(\mathfrak{A}) = \mathrm{Th}_{\forall_0}(\mathfrak{A}, a)_{a \in \mathfrak{A}} \tag{1.4}$$

Similarly, the *n-quantifier diagram* of $\mathfrak{A}$ is the set of sentences

$$n\text{-}\operatorname{diag}(\mathfrak{A}) = \mathrm{Th}_{\exists_n}(\mathfrak{A}, \overline{a})_{\overline{a} \in \mathfrak{A}} \cup \mathrm{Th}_{\forall_n}(\mathfrak{A}, \overline{a})_{\overline{a} \in \mathfrak{A}} \tag{1.5}$$

Finally, the elementary diagram of $\mathfrak{A}$ is $\operatorname{elediag}(\mathfrak{A}) = \bigcup_{n \in \omega} n\text{-}\operatorname{diag}(\mathfrak{A})$.

We will also be interested in infinitary formulae. For a first order language $\mathcal{L}$, we let $\mathcal{L}_{\omega_1,\omega}$ be the closure of $\mathcal{L}$ under countable conjunction and first order operations. We define the syntactic complexity of these formulae as in Ash and Knight [2].

Definition 1.1.3. We let $\Sigma_0 = \Pi_0 = \exists_0$. For any countable ordinal $\alpha \geq 1$, we say that $\varphi \in \mathcal{L}_{\omega_1,\omega}$ is a Σ_α *formula* if $\varphi = \bigvee_{k \in \omega} \exists \overline{x} \psi_k(\overline{x})$, where the ψ_k are all Π_β formulae for $\beta < \alpha$. Similarly, we say that φ is a Π_α *formula* if $\varphi = \bigwedge_{k \in \omega} \forall \overline{x} \psi_k(\overline{x})$, where the ψ_k are all Σ_β formulae.

It is important to note that a Σ_α or Π_α formula φ is not necessarily first order, even when α is finite. For example, the following formula of arithmetic is Σ_1, but not first order.

$$\varphi(x) = \bigvee_{k \in \omega} (x \leq k) \tag{1.6}$$

Unlike first order formulae, infinitary formulae cannot be enumerated, as there are uncountably many of them. To solve this problem, we define an enumeration of the computable infinitary formulae.

Definition 1.1.4. A *code* for a formula $\varphi \in \mathcal{L}_{\omega_1,\omega}$ is a quadruple $(a, c, \overline{x}, e)$ with the following properties.

(a) $a \in O$ is a notation for some ordinal α such that φ is either Σ_α or Π_α.

(b) $c \in \{0, 1\}$ with $c = 0$ if and only if φ is Σ_α.

(c) $\overline{x}$ is a subset of the free variables of φ.

(d) If φ is Σ_α then $e \in \omega$ is such that for all $n \in \omega$, $\{e\}(n) = (b_n, 1, \overline{y}_n, i)$ is a code for a Π_{β_n} formulae ψ_n with

$$\varphi = \bigvee_{n \in \omega} \exists \overline{y}_n \psi_n(\overline{y}_n) \tag{1.7}$$

(e) $e \in \omega$ codes a similar sequence of infinitary formulae when φ is Π_α.

A formula $\varphi \in \mathcal{L}_{\omega_1,\omega}$ is a *computable infinitary formula* if it has a code.

Definition 1.1.4 is the reason why we chose the definition of the hyperarithmetic hierarchy in Definition 1.1.1. It is easy to check that a predicate $\mathcal{R}(f, n)$ is $\Sigma^0_\alpha(\Pi^0_\alpha)$ if and only if it is equivalent in $\mathbb{N}$, the standard model of arithmetic, to a computable $\Sigma_\alpha(\Pi_\alpha)$ formula of second order arithmetic.

Finally, we will take most of our terminology for computable model theory from Harizanov [19]. Unless otherwise noted, we will assume our language $\mathcal{L}$ is computable. This will allow us to measure the complexity of sets of formulae in $\mathcal{L}$.

Definition 1.1.5. Let ϕ_n be a computable enumeration of the formulae in $\mathcal{L}$. For any set $\Phi \subseteq \mathcal{L}$, we say that Φ is *computable* if there is some computable function $f \in 2^\omega$ such that

$$\phi_n \in \Gamma \quad \Leftrightarrow \quad f(n) = 1 \tag{1.8}$$

Similarly, for any class of functions $\Gamma \subseteq 2^\omega$, we say that Γ has complexity Γ if it has function $f \in \Gamma$ satisfying (1.8).

Given this notion of complexity, we can define the fundamental notion of computable model theory, that of a computable structure.

Definition 1.1.6. A structure $\mathfrak{A}$ is *computable* if both the domain of $\mathfrak{A}$ and its atomic diagram $\mathrm{diag}(\mathfrak{A})$ is computable. Furthermore, a computable structure is *n-decidable* if n-$\mathrm{diag}(\mathfrak{A})$ is computable, and *decidable* if $\mathrm{elediag}(\mathfrak{A})$ is computable.

In some cases, we will deal with structures that are not computable. In cases where a structure $\mathfrak{A}$ is not computable, we say that the complexity of $\mathfrak{A}$ is the complexity of $\mathrm{diag}(\mathfrak{A})$ as a subset of $\mathcal{L}$.

In general, we will assume that equality in a computable structure is true equality, and not just another computable relation. This is not a problem, as any computable structure $\mathfrak{A}$ is computably isomorphic to such a structure by enumerating the elements of $\mathfrak{A}$ and letting the domain of the new structure be the first element listed for each equivalence class.

Finally, as structures in model theory are often identified up to isomorphism, we need the following definition to continue this practice in the context of computable model theory.

Definition 1.1.7. An isomorphism from a structure $\mathfrak{A}$ to a computable structure $\mathfrak{B}$ is a *computable presentation of* $\mathfrak{A}$. For convenience, we will often identify the image $\mathfrak{B}$ with the isomorphism itself. If a structure $\mathfrak{A}$ has a computable presentation, it is *computably presentable.*

1.2 Summary of results

This dissertation is essentially divided into two parts. The first part, composed of Chapters 2, 3, and 4 explore characterizations of the first kind for existentially closed models of $\forall_2$ theories. That is, we prove several necessary and sufficient conditions for $\forall_2$ theories to have a computable existentially closed model.

In Chapter 2, we define what it means for a structure to be an existentially closed model of a theory. We introduce many of the more important results about existentially closed models proved in the past few decadess as these will be used to construct computable existentially closed models. We also prove several new classical results about existentially closed models. For example, we show that an existentially closed model $\mathfrak{A}$ of a 2-complete $\forall_2$ theory is algebraically prime if and only if every $\exists$ type realized in $\mathfrak{A}$ is maximal and isolated (See section 2.3 for the appropriate definitions concerning 2-completeness and $\exists$ types). This result answers an open question of Baldwin and Kueker [4].

In Chapter 3, we introduce a general procedure for constructing computable structures. This procedure is an effective version of Robinson forcing, first introduced by Robinson [5]. We use this procedure in Chapter 4 to prove several effective analogues of the results in Chapter 2. In section 4.2, we show that, if we can decide the $\forall_1$ sentences true in a $\forall_2$ theory T, then T has a computable existentially closed model. We extend this result in section 4.3 to prove necessary and sufficient conditions for both computable and 1-decidable existentially closed models of $\forall_2$ theories. We further use these results to show that computable existentially closed models have many properties that are similar to those of decidable models. In particular, section 4.5 and section 4.6 are to devoted to proving analogues of results of Harrington [20] and Millar [36] for computable existentially closed models.

Chapters 5, 6, and 7 of this dissertation are dedicated to characterizations of the second kind. In section 5.2, we show how to measure the algorithmic complexity of a model theoretic property over computable structures. With this definition, we argue that completeness results for model theoretic properties constitute an "anticharacterization". If a model theoretic property $\mathcal{P}$ is complete at its level of complexity, we can dismiss another property $\mathcal{Q}$ as a characterization of $\mathcal{P}$ by measuring the upper bound of its complexity. Furthermore, even if $\mathcal{Q}$ is a characterization of $\mathcal{P}$ in this case, it cannot be any more effective than $\mathcal{P}$, which is one of our primary motivations for computing an upper bound on its complexity.

In sections 5.3–5.5 we report on some joint work with Richard Shore which gives completeness results for several analytic properties. In particular, we show that saturation, rigidity, hyperarithmetic categoricity, and hyperarithmetic stability are all Π^1_1-complete with respect to computable structures. We also show that there is a computable structure $\mathfrak{A}$ such that the class of computable structures isomorphic to $\mathfrak{A}$ is Σ^1_1-complete.

While the bulk of this dissertation is concerned with characterizations for computable structures, the results in sections 5.3–5.5 can be extended to the class of all countable structures. In section 5.6, we show how to define the complexity of model theoretic properties over countable structures in descriptive set theory. With this definition, we prove the appropriate analogues in descriptive set theoretic terms for the results in section 5.3 and section 5.4.

In Chapter 6, we construct a general technique for proving completeness results for several hyperarithmetic properties. For example, we use this technique to show that elementary equivalence is Π^0_ω complete with respect to computable structures. We also use this technique to show that the classes of homogeneous computable structures, atomic computable structures, and computably saturated computable structures are all $\Pi^0_{\omega+2}$ complete. As in Chapter 5, we also prove the analogues for these results in descriptive set theory with respect to countable structures.

Finally, in Chapter 7 we report on joint work with Richard Shore regarding the complexity of computable categoricity. In this chapter, we show that computable categoricity is Π^0_4-hard. We use this result to give a new proof that that the existence of a computable Scott family, a property posed by Goncharov [16] as a characterization of computably categorical structures, is not a necessary condition for computable categoricity.

Chapter 2

Existentially Closed Models

2.1 Introduction

Existentially closed models were first introduced by Rabin [40] and Lindström [34]. They are a natural generalization of algebraic closed fields, and have been quite useful in the field of algebra. They also have several model theoretic properties which make them interesting in their own right. The surveys written by Hodges [25], [26] are valuable references for the study of these models.

As we shall see in Chapter 3, existentially closed models are nice because we can construct them without building their elementary diagram. This will give us a way to build computable models of some theories that have no decidable model. Furthermore, the relationship between existentially closed models and $\exists$-types is often similar to that between models and complete types. As a result, many of the characterizations of decidable models can be adapted to computable existentially closed models. With this in mind, we will define the notion of an existentially closed model and give a survey of some of their more significant properties.

2.2 Existentially closed models and $\forall_2$ theories

A field F is algebraically closed if every system of equations and inequations over F which has a solution in an extension of F already has a solution in F. The property that a system of equations and inequations has a solution is representable by an $\exists$ formula. This fact gives us a natural way to define an existentially closed model.

Definition 2.2.1. Let T be a theory with model $\mathfrak{A} \models T$. $\mathfrak{A}$ is *existentially closed (e.c.) in* T if for every $\exists$ formula $\varphi(\overline{x})$ and $\overline{a} \in \mathfrak{A}$, whenever there is an extension $\mathfrak{A} \subseteq \mathfrak{B} \models T$ such that $\mathfrak{B} \models \varphi(\overline{a})$, then $\mathfrak{A} \models \varphi(\overline{a})$.

In general, it is not so easy to tell whether or not a theory T has an existentially closed model. However, when T is $\forall_2$, it will always have an existentially closed model. In fact, we can say more than this. Just as every field can be extended to an algebraically closed field, so can we extend models of $\forall_2$ theories.

Theorem 2.2.2 (Steinitz [47]). *Let T be an $\forall_2$ theory with $\mathfrak{A} \models T$. Then there is an extension $\mathfrak{A} \subseteq \mathfrak{B} \models T$ such that $\mathfrak{B}$ is e.c. in T.*

The proof of Theorem 2.2.2 is exactly the same as it is for fields. We construct a chain of models $\mathfrak{B}_n \subseteq \mathfrak{B}_{n+1} \models T$ with $\mathfrak{B}_0 = \mathfrak{A}$ such that $\mathfrak{B}_{n+1}$ contains witnesses for each extensional formula $\varphi(\overline{b})$ consistent with $\mathfrak{B}_n$. $\forall_2$ theories are inductive; that is, any union of a chain of models of T is also a model of T. Therefore, $\mathfrak{B} = \bigcup_n \mathfrak{B}_n$ is the desired model.

Most algebraic structures have $\forall_2$ theories. For example, the theory of groups is generated over the signature $\tau = \{\,\cdot, 1\,\}$ by the following list of axioms.

(a) $\forall x, y, z \left((x \cdot y) \cdot z = x \cdot (y \cdot z)\right)$

(b) $\forall x \left(x \cdot 1 = x\right)$

(c) $\forall x \, \exists y \left(x \cdot y = 1\right)$

It is for this reason that existentially closed models are very prevalent in the field of algebra.

An interesting feature of e.c. models of $\forall_2$ theories is that they only depend on the universal sentences true in the theory.

Theorem 2.2.3 (Rabin [40]). *Let T be an $\forall_2$ theory with $\mathfrak{A} \models T$. The following are equivalent.*

(a) $\mathfrak{A}$ *is an e.c. model of* T.

(b) $\mathfrak{A} \models T_\forall$, *and for every* $\exists$ *formula* $\varphi(\overline{x})$ *and* $\overline{a} \in \mathfrak{A}$, *if* $\mathfrak{A} \not\models \varphi(\overline{a})$ *then there is an* $\exists$ *formula* $\psi(\overline{x})$ *such that*

$$\mathfrak{A} \models \psi(\overline{a}) \quad \text{and} \quad \mathfrak{A} \models \forall \overline{x}(\psi(\overline{x}) \to \neg\varphi(\overline{x})) \tag{2.1}$$

(c) $\mathfrak{A}$ *is an e.c. model of* $T_\forall$.

Theorem 2.2.3 provides the foundation for a technique known as Robinson forcing. We will examine Robinson forcing in greater detail in Chapter 3.

2.3 $\exists$-types and existentially saturated models

Models of complete theories are often determined by their complete types. Existentially closed models have a similar relationship with $\exists$-types.

Definition 2.3.1. The $\exists$-*type* of a sequence $\overline{a} \in \mathfrak{A}$ is the set $\Gamma(\overline{x})$ of all $\exists$ formulae $\varphi(\overline{x})$ such that $\mathfrak{A} \models \varphi(\overline{a})$. We denote this set by $\exists$-type$(\overline{a})$. For a theory T, we say that $\Gamma(\overline{x})$ is an $\exists$-*type of* T if $\Gamma = \exists$-type$(\overline{a})$ for some $\overline{a} \in \mathfrak{A} \models T$.

There is one very important difference between $\exists$-types and complete types: for $\exists$-types Γ and Φ, if $\Gamma \subseteq \Phi$, it is not necessarily true that $\Gamma = \Phi$. Therefore, if we want to draw parallels between $\exists$-types and complete types, we need the following definition.

Definition 2.3.2. An $\exists$-type $\Gamma(\overline{x})$ of T is *maximal* if when $\Phi(\overline{x})$ is also an $\exists$-type of T with $\Gamma \subseteq \Phi$, then $\Gamma = \Phi$.

In general, not every $\exists$-type of an $\forall_2$ theory is maximal. For example, consider $(\mathbb{Z}, +)$ and $(\mathbb{Q}, +)$ as groups. Since $\mathbb{Z} \subseteq \mathbb{Q}$, and existential formulae are preserved upwards under inclusion,

$$\exists\text{-type}_{\mathbb{Z}}(1) \subseteq \exists\text{-type}_{\mathbb{Q}}(1) \tag{2.2}$$

However, $\exists y\,(y + y = x)$ is in $\exists$-type$_{\mathbb{Q}}(1)$, but not in $\exists$-type$_{\mathbb{Z}}(1)$. Fortunately, when we restrict ourselves to existentially closed models, this is no longer the case.

Definition 2.3.3. Let $\Psi(\overline{x})$ be a sequence of formulae. We say that Ψ is *realized in* $\mathfrak{A}$ if there is some $\overline{a} \in \mathfrak{A}$ such

$$\varphi(\overline{x}) \in \Psi(\overline{x}) \quad \Rightarrow \quad \mathfrak{A} \models \varphi(\overline{a}) \tag{2.3}$$

Otherwise, we say that $\mathfrak{A}$ *omits* Ψ.

Theorem 2.3.4 (Hodges [25]). *Let T be an $\forall_2$ theory with $\mathfrak{A} \models T$. The following are equivalent.*

(a) $\mathfrak{A}$ is an e.c. model of T.

(b) $\mathfrak{A} \models T_\forall$ and for every $\overline{a} \in \mathfrak{A}$, $\exists$-type$(\overline{a})$ is a maximal $\exists$-type of T.

It is because of Definition 2.3.2 and Theorem 2.3.4 that we say that maximal $\exists$-types in e.c. models are analogous to complete types in other models. An interesting question then is to determine what properties of complete types can be adapted to maximal $\exists$-types in e.c. models. One obvious candidate is saturation.

Definition 2.3.5. Let T be a complete theory with $\mathfrak{A} \models T$ countable. $\mathfrak{A}$ is *saturated for T* if for every $\overline{a} \in \mathfrak{A}$ and every complete type $\Gamma(\overline{x}, \overline{a})$ of $T \cup \mathrm{elediag}(\mathfrak{A})$, Γ is realized in $\mathfrak{A}$.

Theorem 2.3.6 (Morley and Vaught [37]). *Let T be a complete theory with $\mathfrak{A}$, $\mathfrak{B} \models T$ both countable, saturated models. Then $\mathfrak{A} \cong \mathfrak{B}$.*

So any complete theory T has exactly one countable saturated model up to isomorphism. Theorem 2.3.6 is nice because it allows us to treat every countable model of T as a substructure of the unique countable saturated model, when it exists. There is a straightforward analogue of this fact for maximal $\exists$-types.

Definition 2.3.7. Let T be a theory with $\mathfrak{A} \models T$ countable. $\mathfrak{A}$ is *existentially saturated (e.s.) for T* if for every $\overline{a} \in \mathfrak{A}$ and every $\exists$-type $\Gamma(\overline{x}, \overline{a})$ of $T \cup \operatorname{diag}(\mathfrak{A})$, Γ is realized in $\mathfrak{A}$.

Theorem 2.3.8 (Jónsson [27], [28]). *Let T be a theory with $\mathfrak{A}, \mathfrak{B} \models T$ countable existentially saturated structures such that for every existential sentence φ,*

$$\mathfrak{A} \models \varphi \quad \Leftrightarrow \quad \mathfrak{B} \models \varphi \tag{2.4}$$

Then $\mathfrak{A} \cong \mathfrak{B}$.

Unfortunately, even if a theory is $\forall_2$, it does not necessarily have a countable existentially saturated structure. Consider the empty theory T over the signature $\{h^2, S^1, 0\}$. For each $f \in 2^\omega$ define the set of formulae

$$\Gamma_f(x) = \{\, h(S^n(0), x) = S^m(0) \mid f(n) = m \,\} \tag{2.5}$$

For each f, there is some model $\mathfrak{A}$ realizing Γ_f and hence some e.c. model realizing Γ_f. So we can extend each Γ_f to a maximal $\exists$-type Φ_f. It is clear from (2.5) that

$$\Phi_f(x) = \Phi_g \quad \Leftrightarrow \quad f = g \tag{2.6}$$

So there are at least 2^ω maximal $\exists$-types of T. Any countable saturated model must realize them all, which is not possible. For this reason, we have the following theorem.

Theorem 2.3.9 (Pouzet [39]). *Let T be an $\forall_2$ theory. Then T has a countable existentially saturated model if and only if it has countably many maximal $\exists$-types.*

2.4 Prime existentially closed models

As we saw in Theorem 2.3.6, the saturated model of a theory is, in some sense, the largest countable model of the theory. The dual of this notion is that of the elementarily prime model.

Definition 2.4.1. A model $\mathfrak{A} \models T$ is an *elementarily prime model of T* if $\mathfrak{A}$ is elementarily embeddable into all models $\mathfrak{B} \models T$.

Elementary embeddings are exactly those embeddings which preserve the complete types of the structure. Therefore, it reasonable to believe that whether or not a model is elementarily prime should depend on the complete types realized in the model. As the following result demonstrates, this is indeed the case.

Theorem 2.4.2 (Vaught [48]). *Let T be a complete theory with $\mathfrak{A} \models T$ countable. Then $\mathfrak{A}$ is an elementarily prime model of T if and only if every complete type of T realized in $\mathfrak{A}$ is principal.*

Intuitively, there should be a similar result relating embeddings and $\exists$-types. Just as elementary embeddings preserve complete types, so do embeddings preserve the $\exists$-types. Therefore, an interesting question is whether or not there is an analogous result to Theorem 2.4.2 for maximal $\exists$-types and algebraically prime models.

Definition 2.4.3. A model $\mathfrak{A} \models T$ is an *algebraically prime model of* T if $\mathfrak{A}$ is embeddable into all models $\mathfrak{B} \models T$.

Algebraically prime models of $\forall_2$ theories have been studied in great detail by Baldwin and Kueker [4]. In attempting to find a suitable analogue of Theorem 2.4.2, they introduced the concept of an $\exists$-atomic model.

Definition 2.4.4. For any set of formulae $\Psi(\overline{x})$ with free variables among $\overline{x}$, Ψ has $\exists_n(\forall_n)$*-support* if there is some $\exists_n(\forall_n)$ formula $\varphi(\overline{x})$ consistent with T such that for all $\psi(\overline{x}) \in \Psi(\overline{x})$,

$$T \vdash \forall \overline{x} \left(\varphi(\overline{x}) \to \psi(\overline{x}) \right) \tag{2.7}$$

Definition 2.4.5. A structure $\mathfrak{A}$ is a *weakly* $\exists$*-atomic* model of T if $\mathfrak{A} \models T$ and for every $\overline{a} \in \mathfrak{A}$, $\exists$-type$(\overline{a})$ has $\exists$-support. $\mathfrak{A}$ is an $\exists$*-atomic* model of T if every $\exists$-type realized in $\mathfrak{A}$ is maximal and has $\exists$-support.

An $\exists$-atomic model is analogous to an atomic model, or a model whose complete types are all principal. So if we are to continue our analogy between complete types and maximal $\exists$-types, we want a relationship between $\exists$-atomic models and algebraically prime models. As the following result shows, this relationship exists for suitable $\forall_2$ theories.

Definition 2.4.6. A theory T is *n-complete* if for any $\exists_n$ sentence φ, either $T \vdash \varphi$ or $T \vdash \neg\varphi$. An *n-completion* of T is an extension $\widehat{T}$ of T which is n-complete.

Theorem 2.4.7 (Baldwin and Kueker [4]). *Let T be a 1-complete $\forall_2$ theory with $\mathfrak{A} \models T$ countable. Then* (a) $\Rightarrow$ (b) $\Rightarrow$ (c).

(a) $\mathfrak{A}$ is $\exists$-atomic.

(b) $\mathfrak{A}$ is weakly $\exists$-atomic.

(c) $\mathfrak{A}$ is algebraically prime.

Moreover, if $\mathfrak{A}$ is $\exists$-atomic, then $\mathfrak{A}$ is existentially closed.

Unfortunately, this result is not a true analogue of Theorem 2.4.2. We would like to show that, for e.c. models $\mathfrak{A}$, (c) $\Rightarrow$ (a). We can do this if we have a suitable omitting types theorem. Suppose that for any maximal $\exists$-type with $\exists$-support, there is a model $\mathfrak{B} \models T$ which omits this type. Then because embeddings preserve $\exists$-types, if $\mathfrak{A}$ is not $\exists$-atomic, there is some $\mathfrak{B} \models T$ such that $\mathfrak{A}$ is not embeddable in $\mathfrak{B}$. Provided that an $\forall_2$ theory is 2-complete, we have such an omitting types theorem.

Theorem 2.4.8 (Omitting $\forall_2$-types Theorem). *Let T be a consistent $\forall_2$ theory which is 2-complete. For each $n < \omega$, let $\Psi_n(\overline{x})$ be a set of $\forall_2$ formulae with no $\exists$-support over T. Then T has a model omitting all of the Ψ_n.*

Theorem 2.4.8 merits some discussion in its own right. Therefore, we will defer its proof until section 2.5. Instead, we will use it to prove our analogue of Theorem 2.4.2.

Lemma 2.4.9. *Let T be a 2-complete $\forall_2$ theory with $\mathfrak{A} \models T$ countable. If $\mathfrak{A}$ is an algebraically prime model of T then $\mathfrak{A}$ is weakly $\exists$-atomic.*

Proof. Suppose for a contradiction that there is some $\overline{a} \in \mathfrak{A}$ such that $\exists$-type$(\overline{a})$ has no $\exists$-support. By Theorem 2.4.8, there is some $\mathfrak{B} \models T$ which omits $\exists$-type$(\overline{a})$. As existential formulae are preserved upwards under embeddings, there is no embedding from $\mathfrak{A}$ to $\mathfrak{B}$. This contradicts the fact that $\mathfrak{A}$ is algebraically prime. □

Theorem 2.4.10. *Let T be a 2-complete $\forall_2$ theory with $\mathfrak{A} \models T$ countable. Then $\mathfrak{A}$ is an existentially closed algebraically prime model of T if and only if it is $\exists$-atomic.*

Proof. Suppose $\mathfrak{A} \models T$ is countable and $\exists$-atomic. The forward direction is just Theorem 2.4.7. For the other direction, we appeal to Lemma 2.4.9. Take any $\overline{a} \in \mathfrak{A}$. As $\mathfrak{A}$ is algebraically prime, $\exists$-type$(\overline{a})$ is supported by some $\exists$ formula φ. As $\mathfrak{A}$ is existentially closed, $\exists$-type$(\overline{a})$ is maximal and so $\varphi \in \exists$-type$(\overline{a})$. Furthermore, by maximality,

$$T \vdash \forall\overline{x}\,\big(\varphi(\overline{x}) \to \psi(\overline{x})\big) \quad \Rightarrow \quad \psi \in \exists\text{-type}(\overline{a}) \tag{2.8}$$

So φ supports $\exists$-type$(\overline{a})$. □

Theorem 2.4.7 only requires that the theory T be 1-complete. Therefore, it would be nice if we could extend Theorem 2.4.10 to 1-complete $\forall_2$ theories. Unfortunately, this is not possible.

Example 2.4.11. There is a consistent, 1-complete $\forall_2$ theory with an existentially closed algebraically prime model $\mathfrak{A}$ which is not $\exists$-atomic.

Proof. We will use the following construction from Baldwin and Kueker [4]. T will be a theory of the signature $\tau = \{E\}$ where E is a binary relation. T will be the theory induced by the following list of axioms.

(a) E is an equivalence relation.

(b) The are infinitely many equivalence classes. That is,

$$\exists x_1, \ldots, x_n \bigwedge_{i \neq j} (\neg x_i \; E \; x_j) \tag{2.9}$$

for all n.

(c) There is only one finite equivalence class. That is,

$$\forall x, y\, \big(\big(\exists^{\leq n} z\, (x\ E\ z) \wedge \neg x\ E\ y\big) \rightarrow \exists^{\geq n+1} w\, (x\ E\ z)\big) \tag{2.10}$$

for all n.

This is a consistent $\forall_2$ theory. Its models are just the structures $\mathfrak{A}_n$, $n \geq 1$, which have one equivalence class of n elements, and the structure $\mathfrak{A}_0$ which has no finite equivalence class. $\mathfrak{A}_i \subseteq \mathfrak{A}_j$ for all i, j. Since $\exists$ sentences are preserved upwards under inclusion, $\mathrm{Th}_\exists(\mathfrak{A}_i) = \mathrm{Th}_\exists(\mathfrak{A}_j)$ for all i, j. As these are the only models for T, T is 1-complete.

The model $\mathfrak{A}_0$ with no finite equivalence classes is existentially closed. Take any $\overline{a} \in \mathfrak{A}_0$ and $\exists$ formula φ such that $\mathfrak{B} \models \varphi(\overline{a})$ for some $\mathfrak{B} \supseteq \mathfrak{A}_0$. As $\varphi(x)$ is existential and relational, there is some finite substructure $\mathfrak{C} \subseteq \mathfrak{B}$ with $\mathfrak{C} \models \varphi(a)$; $\mathfrak{C}$ is just enough of $\mathfrak{B}$ to provide witnesses for φ. As all of the equivalence classes in $\mathfrak{C}$ are finite, there is an embedding $f : \mathfrak{C} \hookrightarrow \mathfrak{A}_0$ which is the identity on $\overline{a}$. So $\mathfrak{A}_0 \models \varphi(\overline{a})$.

While $\mathfrak{A}_0$ is $\exists$-atomic over $\mathrm{Th}_{\forall_2}(\mathfrak{A}_0)$, it is not $\exists$-atomic over T. Take any $a \in \mathfrak{A}_0$ and let $\Gamma = \exists$-type(a). Note then that

$$\big\{\, \exists^{\geq n} y\, (x\ E\ y) \mid n \in \omega \,\big\} \subseteq \Gamma(x) \tag{2.11}$$

Suppose for a contradiction that $\varphi(x)$ is an $\exists$-support for Γ. By maximality, $\mathfrak{A}_0 \models \varphi(a)$. As before, there is some finite substructure $\mathfrak{B} \subseteq \mathfrak{A}_0$ with $\mathfrak{B} \models \varphi(a)$.

Let n be the number of elements in the equivalence class of a in $\mathfrak{B}$. Then we can map $\mathfrak{B}$ into $\mathfrak{A}_n$ by an embedding which takes a to a_n, where a_n is an element of the finite equivalence class of $\mathfrak{A}_n$. Hence $\mathfrak{A}_n \models \varphi(a_n)$. As φ supports Γ, we have that $\mathfrak{A}_n \models \Gamma(a_n)$. However, $\mathfrak{A}_n \not\models \exists^{\geq n+1} y\, (a_n\ E\ y)$, which is a contradiction. □

Therefore, this characterization is only valid for 2-complete $\forall_2$ theories. However, it does give us a nice way to characterize when a theory of this type has an e.c. algebraically prime model. For this we need to introduce the notion of an isolated, maximal $\exists$-type.

Definition 2.4.12. For any maximal $\exists$-type $\Gamma(\overline{x})$ over T, with free variables among $\overline{x}$, Γ is *isolated* if there is some formula $\varphi \in \Gamma$ such that Γ is the unique maximal $\exists$-type over T containing φ. In this case we say that φ *isolates* Γ.

It is easy to see that if a maximal $\exists$-type has $\exists$-support, then it is isolated by this support. However, the converse is not true in general. If a maximal $\exists$-type has $\exists$-support φ, then this is the only $\exists$-type, maximal or otherwise, containing φ. As illustrated in Baldwin and Kueker [4], even though an $\exists$ formula $\varphi(seq)$ may extend to a unique maximal $\exists$-type, there can be several nonmaximal $\exists$-types containing φ.

Fortunately, this cannot happen for 2-complete $\forall$ theories.

Proposition 2.4.13. *Let T be a 2-complete $\forall_2$ theory. Then every isolated, maximal $\exists$-type over T has $\exists$-support.*

Proof. Let $\Gamma(\overline{x})$ be a maximal $\exists$-type isolated by the formula $\varphi \in tpform$. It is enough to show that for any e.c. model $\mathfrak{A} \models T$,

$$\mathfrak{A} \models \forall \overline{x} \left(\varphi(\overline{x}) \rightarrow \psi(\overline{x})\right) \text{ for all } \psi \in \Gamma \tag{2.12}$$

The sentences in (2.12) are all $\forall_2$, so in this case (2.7) in Definition 2.4.4 would follow from the 2-completeness of T.

Let $\mathfrak{A} \models T$ be existentially closed. Take any $\overline{a} \in \mathfrak{A}$ such that $\mathfrak{A} \models \varphi(\overline{a})$. As every type realized in $\mathfrak{A}$ is maximal, $\exists\text{-type}(\overline{a}) = \Gamma$. Therefore, $\mathfrak{A} \models \psi(\overline{a})$ for all $\psi \in \Gamma$. As $\overline{a} \in \mathfrak{A}$, we are done. □

Theorem 2.4.14. *Let T be a 2-complete $\forall_2$ theory. Then T has an e.c. algebraically prime model if and only if every $\exists$-formula consistent with T is contained in an isolated, maximal $\exists$-type over T.*

Proof. ($\Rightarrow$) Let $\mathfrak{A}$ be an e.c. algebraically prime model of T with $\varphi(\overline{x})$ be an $\exists$ formula consistent with T. As T is 2-complete, there is some $\overline{a} \in \mathfrak{A}$ with $\mathfrak{A} \models \varphi(\overline{a})$. By Theorem 2.4.10, $\exists\text{-type}(\overline{a})$ is maximal and has $\exists$-support. So $\exists\text{-type}(\overline{a})$ is an isolated maximal $\exists$-type containg φ.

($\Leftarrow$) Suppose every $\exists$ formula consistent with T is contained in an isolated maximal $\exists$-type over T. By a result in Chapter 3 of Hodges [25], there is some $\mathfrak{A} \models T$ such that every $\exists$-type realized in $\mathfrak{A}$ is maximal and isolated over T. As T is 2-complete, every $\exists$-type realized in $\mathfrak{A}$ has $\exists$-support, and hence $\mathfrak{A}$ is $\exists$-atomic. So $\mathfrak{A}$ is algebraically prime by Theorem 2.4.10. □

When T is a complete $\forall_2$ theory, we can get an even better analogue of Theorem 2.4.2. That is because, in this case, every isolated maximal $\exists$-type extends to a unique complete type.

Lemma 2.4.15. *Let T be a complete theory with $\mathfrak{A} \models T$ countable. The following are equivalent.*

(a) *Every complete type realized in $\mathfrak{A}$ is principally generated over T by an $\exists$ formula.*

(b) *$\mathfrak{A}$ is an $\exists$-atomic model of T.*

(c) *For every $\overline{a} \in \mathfrak{A}$ there is some $\exists$ formula $\varphi_{\overline{a}}$ such that $\mathfrak{A} \models \varphi_{\overline{a}}(\overline{b})$ if and only if $(\mathfrak{A}, \overline{a}) \cong (\mathfrak{A}, \overline{b})$.*

Proof. (a) $\Rightarrow$ (b) This case is obvious.

(b) $\Rightarrow$ (c) Given $\overline{a} \in \mathfrak{A}$, let $\varphi_{\overline{a}}$ be the support for $\exists$-type$(\overline{a})$. Then, for all $\exists$-formulae ψ,

$$\psi(\overline{a}) \in \exists\text{-type}(\overline{a}) \quad \Leftrightarrow \quad T \vdash \forall \overline{x}\, \big(\varphi_{\overline{a}}(\overline{x}) \to \psi(\overline{x})\big) \tag{2.13}$$

We show by a back-and-forth argument that $\varphi_{\overline{a}}$ determines $\overline{a}$ up to automorphism. Suppose that $\overline{b} \in \mathfrak{A}$ also satisfies $\varphi_{\overline{a}}$. We extend $\overline{a} \mapsto \overline{b}$ to an automorphism of $\mathfrak{A}$ via a back-and-forth argument. We will build a bijection f on $\mathfrak{A}$ in stages such that for all s

$$\mathfrak{A} \models \varphi_{\mathrm{dom}(f_s)}\big(\,\mathrm{rng}(f_s)\big) \tag{2.14}$$

Stage $s = 0$:
Initially, we set $f_0 = (\overline{a} \mapsto \overline{b})$.

Stage $s + 1 = 2e$:
Let $u \in \mathfrak{A}$ be the least element not in $\mathrm{dom}(f_s)$. Since

$$\exists y\, \varphi_{\mathrm{dom}(f_s),u}(\overline{x}, y) \in \exists\text{-type}\,\big(\,\mathrm{dom}(f_s)/\overline{d}\big) \tag{2.15}$$

the property (2.14) implies that

$$\mathfrak{A} \models \exists y\, \varphi_{\mathrm{dom}(f_s),u}\big(\,\mathrm{rng}(f_{s+1}, y)\big) \tag{2.16}$$

Let $f(u)$ be the least y satisfying (2.16). !

Stage $s + 1 = 2e + 1$:
This stage is analogous to the stage $s + 1 = 2e$ except that now $u \in \mathfrak{A}$ is the least element not in $\mathrm{rng}(f_s)$.

It is clear that $f = \lim_s f_s$ is a bijection on $\mathfrak{A}$. To see that it is an automorphism, take any $\overline{p} \in \mathfrak{A}$. Let s be the least stage such that $\overline{p} \in \mathrm{dom}(f_s)$. By padding free variables, $\varphi_{\overline{p}} \in \exists$-type $\big(\,\mathrm{dom}(f_s)/\overline{d}\big)$. So, (2.14) implies that $\mathfrak{A} \models \varphi_{\overline{p}}\big(f(\overline{p})\big)$. Now take any atomic formula $\xi(\overline{x})$. Clearly,

$$\mathfrak{A} \models \xi(\overline{p}) \quad \Leftrightarrow \quad \xi(\overline{x}) \in \exists\text{-type}(\overline{p}/\overline{d}) \tag{2.17}$$

Hence $\mathfrak{A} \models \xi(\overline{p})$ if and only if $\mathfrak{A} \models \xi(\big(f(\overline{p})\big)$.

(c) $\Rightarrow$ (a) For $\overline{a} \in \mathfrak{A}$, let $\Phi_{\overline{a}}$ be the complete type of $\overline{a}$ in $\mathfrak{A}$. Suppose $\psi \in \Phi_{\overline{a}}$. Then $\mathfrak{A} \models \Phi(\overline{b})$ for any $\overline{b}$ in the automorphism orbit of $\overline{a}$. Therefore, since $\varphi_{\overline{a}}$ identifies $\overline{a}$ up to automorphism,

$$\mathfrak{A} \models \forall \overline{x}\, \big(\varphi_{\overline{a}}(\overline{x}) \to \psi(\overline{x})\big) \tag{2.18}$$

As T is complete

$$T \vdash \forall \overline{x}\, \big(\varphi_{\overline{a}}(\overline{x}) \to \psi(\overline{x})\big) \tag{2.19}$$

$\square$

Given this lemma, we can now replace "existentially closed and algebraically prime" in Theorem 2.4.10 with "elementarily prime".

Theorem 2.4.16. *Let T be a complete $\forall_2$ theory with $\mathfrak{A} \models T$. The following are equivalent.*

(a) $\mathfrak{A}$ is an elementarily prime model of T.

(b) $\mathfrak{A}$ is an existentially closed, algebraically prime model of T.

(c) $\mathfrak{A}$ is an $\exists$-atomic model of T.

Proof. (a) $\Rightarrow$ (b) As $\mathfrak{A}$ is elementarily prime, we need only show that $\mathfrak{A}$ is existentially closed in T. By Theorem 2.2.2, there is an extension $\mathfrak{A} \subseteq \mathfrak{B} \models T$ such that $\mathfrak{B}$ is existentially closed. Since $\mathfrak{A}$ is elementarily prime, there is a $\mathfrak{C} \cong \mathfrak{B}$ such that $\mathfrak{A} \preceq \mathfrak{C}$. $\mathfrak{C}$ is e.c. since $\mathfrak{B}$ is. So, by a result in Hodges [26], $\mathfrak{A}$ is e.c.

(b) $\Rightarrow$ (c) This direction is just Theorem 2.4.10.

(c) $\Rightarrow$ (a) We see from Lemma 2.4.15 that $\mathfrak{A}$ is atomic, so we are done. $\square$

Theorem 2.4.16 is important for two reasons. First, it reinforces our claim that maximal $\exists$-types of existentially closed models are very similar to complete types. Secondly, it gives us a way to characterize when $\forall_2$ theories have elementarily prime models. The following result will be useful in Chapter 3 when we characterize $\forall_2$ theories with computable elementarily prime models.

Corollary 2.4.17. *Let T be a complete $\forall_2$ theory. Then T has an elementarily prime model if and only if every $\exists$ formula consistent with T is contained in an isolated, maximal $\exists$-type over T.*

Proof. This is immediate from Theorem 2.4.14 and Theorem 2.4.16. $\square$

An interesting open question with regards to Theorem 2.4.16 is whether we can replace the completeness hypothesis with n-completeness for some $n \in \omega$. As the following example of Baldwin and Kueker [4] shows, 1-completeness is not sufficient.

Example 2.4.18 (Baldwin and Kueker [4]). There is a 1-complete $\forall_2$ theory with an e.c. algebraically prime model that is not $\exists$-atomic.

Therefore, 2-completeness is necessary for Theorem 2.4.10, and hence Theorem 2.4.16. However, there is no known example showing that completeness is necessary for Theorem 2.4.16. In particular, it is still unknown whether there is a 2-complete $\forall_2$ theory with an e.c. algebraically prime model which is not elementarily prime.

2.5 Omitting $\exists$-types

We are still left with the task of proving Theorem 2.4.8. In order to do this, we need the following result for sets of universal formulae.

Definition 2.5.1. A set of formulae $\Phi(\overline{x})$ is an $\exists_n(\forall_n)$-set if all of its elements are $\exists_n(\forall_n)$ formulae.

Theorem 2.5.2 (Keisler [30]). *Let T be a consistent $\forall_2$ theory. For each $n < \omega$, let $\Psi_n(\overline{x})$ be a set of universal formulae with no $\exists$-support over T. Then T has an existentially closed model omitting all of the Ψ_n.*

We can extend this result to sets of $\forall_2$ formulae by a process known as relativization.

Proposition 2.5.3. *Let T be a consistent $\forall_3$ theory. For each $n < \omega$, let $\Psi_n(\overline{x})$ be an $\forall_2$-set with no $\exists_2$-support over T. Then T has a model which omits all of the Ψ_n.*

Proof. We introduce a new relation R_φ for each $\exists$ formula $\varphi(\overline{x}) \in \mathcal{L}$. Let $\widehat{T}$ be the theory in $\mathcal{L}(R_\varphi)_\varphi$ given by the following list of axioms.

(a) $\forall \overline{x}\big(\varphi(\overline{x}) \leftrightarrow R_\varphi(\overline{x})\big)$ if φ is an $\exists$ formula.

(b) $\forall \overline{x}\, \exists \overline{y}\, \neg R_{\neg\varphi(\overline{x},\overline{y})}(\overline{x}, \overline{y})$ for all $\forall_3$ sentences $\forall \overline{x}\, \exists \overline{y}\, \varphi(\overline{x}, \overline{y}) \in T$.

$\widehat{T}$ is a extension of T in $\mathcal{L}(R_\varphi)_\varphi$, since $\widehat{T} \vdash \psi$ for all $\forall_3$ sentences $\psi \in T$. $\widehat{T}$ is $\forall_2$, since we only introduced relations R_φ for $\exists$ formulae $\varphi(\overline{x})$. Finally, $\widehat{T}$ is consistent. Take any model $\mathfrak{A} \vDash T$. We can expand $\mathfrak{A}$ to a structure $\widehat{\mathfrak{A}}$ in $\mathcal{L}(R_\varphi)$ by defining

$$\widehat{\mathfrak{A}} \vDash R_\varphi(\overline{a}) \quad \Leftrightarrow \quad \mathfrak{A} \vDash \varphi(\overline{a}) \tag{2.20}$$

As $\widehat{\mathfrak{A}}$ is an expansion of $\mathfrak{A}$, $\widehat{\mathfrak{A}} \vDash T$. So it is clear that $\widehat{\mathfrak{A}} \vDash \widehat{T}$.

Every $\forall_2$ formula in $\mathcal{L}$ is equivalent over $\widehat{T}$ to an $\forall$ formula in $\mathcal{L}(R_\varphi)$. In particular, if $\psi(\overline{x}, \overline{y})$ is an $\forall$ formula, then

$$\widehat{T} \vdash \forall \overline{x}\big(\forall \overline{y} R_\psi(\overline{x}, \overline{y}) \leftrightarrow \forall \overline{y} \psi(\overline{x}, \overline{y})\big) \tag{2.21}$$

So for each n there is a set $\Phi_n(\overline{x})$ of $\forall$ formulae which is equivalent over $\widehat{T}$ to $\Psi_n(\overline{x})$. Moreover, by a simple induction on the complexity of formulae in $\mathcal{L}(R_\varphi)$, any $\exists$ formula of $\mathcal{L}(R_\varphi)$ is equivalent to an $\exists_2$ formula of $\mathcal{L}$. Therefore, since Ψ_n is not supported by an $\exists_2$ formula, Φ_n is not supported by an $\exists$ formula.

Applying Theorem 2.5.2, we see that $\widehat{T}$ has a model $\widehat{\mathfrak{A}}$ which omits all of the Ψ_n. If we let $\mathfrak{A}$ be the reduction of $\widehat{\mathfrak{A}}$ to $\mathcal{L}$, then $\mathfrak{A} \vDash T$. Moreover, $\mathfrak{A}$ omits all of the Ψ_n since $\widehat{\mathfrak{A}}$ does. $\square$

This proof is a rather nice way to generalize constructions of models of $\forall_2$ theories to constructions of models of more complicated theories. We will return to this construction when we build computable models of theories in Chapter 4.

One problem with this construction, however, is that we can no longer guarantee that $\mathfrak{A}$ is existentially closed in T, even if T is $\forall_2$. Even though $\widehat{\mathfrak{A}}$ is e.c. in $\widehat{T}$, this does not imply that $\mathfrak{A}$ is e.c. in T. In fact, Proposition 2.5.3 is false if we insist that $\mathfrak{A}$ be e.c.

Example 2.5.4. There is a consistent $\forall_2$ theory T and set $\Psi(x)$ of existential formulae with no $\exists_2$-support over T such that Ψ is realized in any existentially closed model of T.

Proof. Let T be the theory of chains over a binary relation R. That is, T is given by the following axioms.

(a) R is a directed graph.

(b) Every element has at most one predecessor.

(c) Every element has at most one successor.

It is clear that T is consistent and $\forall_2$. As in Example 2.4.11, let Ψ be the set of existential formulae

$$\Psi(x) = \left\{ \exists^{\geq n} y \, (x \; R \; y) \mid n \in \omega \right\} \tag{2.22}$$

We will show that Ψ is our desired set of formulae.

Let $\mathfrak{A}$ be the model of T with exactly one chain of length n for each n. Then $\mathfrak{A}$ omits Ψ By compactness, there is a structure $\mathfrak{A} \preceq \mathfrak{B}$ such that $\mathfrak{B}$ has a chain with no final element. So $\mathfrak{B}$ realizes Ψ. Let b be any element of $\mathfrak{B}$ realizing Ψ and let φ be any $\exists_2$ formula such that $\mathfrak{B} \models \varphi(b)$. As $\mathfrak{B} \models \exists x \varphi(x)$ and $\mathfrak{B} \equiv \mathfrak{A}$, there is some element $a \in \mathfrak{A}$ such that $\mathfrak{A} \models \varphi(a)$. However, a does not realize Ψ, so φ cannot support Ψ over T. Hence Ψ has no $\exists_2$-support over T.

Finally, suppose for a contradiction that there is some e.c. $\mathfrak{A} \models T$ such that $\mathfrak{A}$ omits Ψ. Take any $a \in \mathfrak{A}$ and let n be least such that $\mathfrak{A} \not\models \exists^{\geq n} y \, (a \; R \; y)$. We extend $\mathfrak{A}$ to a $\mathfrak{B} \models T$ by adding one more element to the end of the chain containing a. Then $\mathfrak{B} \models \exists^{\geq n} y \, (a \; R \; y)$ which contradicts the fact that $\mathfrak{A}$ was existentially closed. □

Neither of these issues presented a problem in our proof of Theorem 2.4.10. We did not need to know that the omitting model is existentially closed; we only needed to know that the algebraically prime model is existentially closed.

However we did need to know that we can omit an $\exists$-type whenever it does not have $\exists$-support. This is the primary difference between Proposition 2.5.3 and Theorem 2.4.8, and it is an important one. As we saw in Example 2.4.11, a maximal $\exists$-type can have $\exists_2$-support even if it does not have $\exists$-support. The type

Ψ in (2.11) has no $\exists$-support but it cannot be omitted in any model of T, so it must have $\exists_2$-support.

Fortunately, this can never happen when T is 2-complete. The following result, together with Proposition 2.5.3, gives us Theorem 2.4.8.

Proposition 2.5.5. *Let T be a consistent, 2-complete $\forall_2$ theory. Let $\Phi(\overline{x})$ be any set of formulae with no $\exists$-support over T. Then $\Phi(\overline{x})$ has no $\exists_2$-support over T.*

Proof. Suppose for a contradiction that $\varphi(\overline{x})$ is an $\exists_2$ formula supporting $\Phi(\overline{x})$ over T. Then let

$$\varphi(\overline{x}) = \exists \overline{y}\, \psi(\overline{x}, \overline{y}) \tag{2.23}$$

where ψ is universal. The set $\{\psi(\overline{x}, \overline{y})\}$ has no $\exists$-support over T; otherwise $\Phi(\overline{x})$ would also have $\exists$-support. By Theorem 2.5.2, there is some model of T omitting $\psi(\overline{x}, \overline{y})$. Hence $T \cup \{\forall \overline{x}\, \neg\varphi(\overline{x})\}$ is consistent. As T is 2-complete, we have that $T \vdash \forall \overline{x}\, \neg\varphi(\overline{x})$, contradicting our assumption that φ was a support of Φ. $\square$

It is a good idea to point out why 2-completeness is necessary in Theorem 2.4.8. The technique employed in Proposition 2.5.5 suggests that we should be able to generalize Theorem 2.4.8 to all $\forall_2$ theories. If an $\forall_2$-set Φ has $\exists_2$-support φ, but no $\exists$-support, it seems that we should be able to extend T to a new theory

$$T^+ = T \cup \{\, \forall \overline{x} \neg\varphi(\overline{x}) \mid \varphi \text{ is a support of } \Phi \,\} \tag{2.24}$$

and apply Proposition 2.5.3.

Unfortunately, when we extend a theory, there is the possibility that we may introduce new supports for our set Φ. Moreover, these new supports could be $\exists$ formulae, so we cannot simply apply Proposition 2.5.5 again. This problem is exactly what happens in Example 2.4.11. The $\exists_2$-supports of Ψ in (2.11) are formulae of the form

$$\varphi_n(x) = \exists y\, \exists^{\leq n} z\, \big(\neg x\ E\ y \wedge y\ E\ z\big) \tag{2.25}$$

So, if we construct T^+ as in (2.24), we get the complete theory of $\mathfrak{A}_0$, the model with no finite equivalence classes. In this theory, Ψ is supported by the formula $(x = x)$. So 2-completeness is necessary in Theorem 2.4.8.

Chapter 3

Effective Model Theoretic Forcing

3.1 Introduction

A standard type of characterization in model theory is one that equates a syntactic property with a semantic one. A typical example is the completeness theorem, first proved by Gödel [14], which states that a theory has a model if and only if it is consistent. Consistency is a proof theoretic or syntactic property, while the existence of a model is a property about the semantics of our language.

This type of characterization exists in computable model theory as well. In particular, Henkin's [23] proof of the completeness theorem characterizes when a theory has a decidable model.

Theorem 3.1.1 (Henkin [23]). *A theory T has a decidable model if and only if it has a consistent computable completion.*

While this is a nice result, it does not characterize when a theory has a computable model. We know from the work of Gödel [15] that Peano arithmetic has no decidable completion, but it does have a computable model, namely the standard model of arithmetic. Therefore, we would like to establish a characterization similar to Theorem 3.1.1 of theories with computable models.

This is a difficult problem because we generally construct a model of a theory by a constructing the elementary diagram of the model. If we were to make this process effective, then the elementary diagram would be computable and hence the model would be decidable. Therefore, if we want an analogue of Theorem 3.1.1 for the existence of computable models, we need a way to construct computable models that does not necessarily make the elementary diagram computable.

One possibility is to construct only the atomic diagram of the model, or even some large part of the elementary diagram. If we take this approach, then we need some way to ensure that the structure we construct is indeed a model of our theory, even when the part of the elementary diagram that we construct does not contain all of the consequences of the theory. This problem is the motivation behind model theoretic forcing.

In this chapter, we will introduce the terminology and basic results of model theoretic forcing. Then we will show how model theoretic forcing can be done effectively, and how it can be used to construct computable models. These results will allow us to prove analogues of Theorem 3.1.1 for $\forall_2$ theories in Chapter 4.

3.2 Model theoretic forcing

Model theoretic forcing was first introduced by Robinson [5]. It is a generalization of the work of Cohen [9] for the construction of models of arbitrary theories. The work by Hodges [25] is an excellent survey of the work done in model theoretic forcing. In particular, our notation in this section is taken from this work.

While we will be interested in model theoretic forcing largely in the context of $\forall_2$ theories, it does not need to be so restricted. It can be applied to theories of any complexity, provided that we have a suitable notion of consistency.

Definition 3.2.1. Let $\mathcal{L}$ be a countable language. A *notion of consistency* for $\mathcal{L}$ is a set N of subsets of $\mathcal{L}$ such that the following all hold.

(a) For every $p \in N$ and every closed term τ, $p \cup \{ (\tau = \tau) \} \in N$.

(b) For every $p \in N$, atomic formula ξ, and closed terms τ, σ, if $\xi(\sigma) \in p$ and p contains either $(\sigma = \tau)$ or $(\tau = \sigma)$, then $p \cup \{ \xi(\tau) \} \in N$.

(c) For every $p \in N$ and every atomic sentence ξ, ξ and $\neg\xi$ are not both in p.

(d) For every $p \in N$, if $\neg\neg\varphi \in p$, then $p \cup \{ \varphi \} \in N$.

(e) For every $p \in N$, if $\varphi \wedge \psi \in p$, then $p \cup \{ \varphi, \psi \} \in N$.

(f) For every $p \in N$, if $\neg(\varphi \wedge \psi) \in p$, then either $p \cup \{ \neg\varphi \} \in N$ or $p \cup \{ \neg\psi \} \in N$.

(g) For every $p \in N$, if $\exists x \varphi(x) \in p$, then $p \cup \{ \varphi(\tau) \} \in N$ for some closed term $\tau \in \mathcal{L}$.

(h) For every $p \in N$, if $\neg\exists x \varphi(x) \in p$, then $p \cup \{ \neg\varphi(\tau) \} \in N$ for all closed terms $\tau \in \mathcal{L}$.

Elements of N are called *N-conditions*, or just *conditions* when N is clear.

The name comes from the fact that these sets N are often of the form

$$N_T^{\mathrm{con}} = \{ p \subseteq \mathcal{L} \mid p \text{ is finite and } T \cup p \text{ is consistent.} \} \tag{3.1}$$

Given this set, we define a game between two players player$_\forall$ and player$_\exists$. The players alternate moves, building an increasing chain $\langle p_s \rangle_{s \in \omega}$ of conditions. By compactness, the union of these conditions $p_\omega = \bigcup_{s \in \omega} p_s$ is a consistent set of sentences in $\mathcal{L}$. Player$_\exists$ plays to ensure that p_ω is a complete extension of T with some additional properties. Player$_\forall$ tries to prevent player$_\exists$ from succeeding.

We can modify this idea to a game where player$_\exists$ plays to build a model of T instead of a complete extension. We can use the atomic sentences in p_ω to define the diagram of a structure of $\mathcal{L}$. To do this, we extend p_ω to an $=$-closed set of sentences.

Definition 3.2.2. A set U of sentences is *$=$-closed* if it has the following properties.

(a) For every closed term of τ, the atomic sentence $(\tau = \tau)$ is in U.

(b) If $\xi(x)$ is an atomic formula of $\mathcal{L}$ and the equation $(\sigma = \tau)$ is in U, then $\xi(\sigma) \in U$ if and only if $\xi(\tau) \in U$.

Proposition 3.2.3 (Hodges [25]). *Let U be an $=$-closed set of atomic sentences in $\mathcal{L}$. Then there is a least structure $\mathfrak{A}(U)$, called the* canonical model of U *satisfying exactly the atomic sentences of U. Furthermore, every element $a \in \mathfrak{A}(U)$ is of the form $\tau^{\mathfrak{A}(U)}$ for some closed term τ of $\mathcal{L}$.*

For a given play $\langle p_s \rangle$, p_ω may not be $=$-closed. However, p_ω is contained in an $=$-closed set; we just extend p_ω to a maximally consistent set of atomics sentences in $\mathcal{L}$. Furthermore, it is clear from Definition 3.2.2 that the intersection of all $=$-closed extensions of p_ω is also $=$-closed. This is the least such $=$-closed extension of p_ω, which we denote $\overline{p}_\omega$. By Proposition 3.2.3, there is a least structure satisfying exactly the atomic sentences of $\overline{p}_\omega$. Since this structure is uniquely determined by p_ω, we denote it by $\overline{\mathfrak{A}}(p_\omega)$, or just $\overline{\mathfrak{A}}$ when p_ω is clear. We call $\overline{\mathfrak{A}}(p_\omega)$ the *compiled structure* for p_ω.

In this new game, player$_\exists$ plays to ensure that $\overline{\mathfrak{A}}(p_\omega)$ has certain properties, while player$_\forall$ tries to prevent player$_\exists$ from succeeding. The problem with this approach is that, whenever $\mathcal{L}$ has no constant symbols, there are no closed terms in $\mathcal{L}$. In this case, $\overline{\mathfrak{A}}(p_\omega)$ is the empty structure. We can solve this problem by extending Definition 3.2.1 to include an infinite set of new constant symbols.

Definition 3.2.4. Let $\mathcal{L}$ be a countable language with W a countable set of constants not appearing in $\mathcal{L}$. A *notion of forcing* for $\mathcal{L}$ over W is a notion of consistency N for $\mathcal{L}(W)$ such that

(a) If $p \in N$, τ is a closed term of $\mathcal{L}(W)$ and $c \in W$ occurs nowhere in p or τ, then $p \cup \{\tau = c\} \in N$.

(b) At most finitely many elements of W occur in any $p \in N$.

We call W the set of *witnesses* for N.

Given a notion of forcing N, we can now define a game between player$_\forall$ and player$_\exists$. Let $\mathcal{P}$ be some model theoretic property. We start with the condition $p_0 = \emptyset$, and let X be an infinite and co-infinite subset of ω. At each stage s, the players chose some $p_s \in N$ such that $p_{s-1} \subseteq p_s$; player$_\forall$ chooses p_s at stages $s \notin X$

while player$_\exists$ chooses it at stages $s \in X$. At the end of the game we have an increasing chain $\langle p_s \rangle$ which we call the *play* of the game. Player$_\exists$ wins the game if $\overline{\mathfrak{A}}(p_\omega)$ has property $\mathcal{P}$; otherwise player$_\forall$ wins. If player$_\exists$ wins the game with a play $\langle p_s \rangle$, we call it a winning play.

We denote this game as $G_N(\mathcal{P}; X)$, or just $G(\mathcal{P}; X)$ when N is clear. We are interested in those properties for which player$_\exists$ can win $G(\mathcal{P})$ no matter what player$_\forall$ does. This idea is made more clear by the following definition.

Definition 3.2.5. A *winning strategy* for $G(\mathcal{P}; X)$ is a function $f : N^{<\omega} \to N$ such that any play $\langle p_n \rangle_{n \in \omega}$ with

$$p_s = f(\langle p_0, \dots, p_{s-1}) \quad \text{for all } s \in X \tag{3.2}$$

is a winning play for $G(\mathcal{P}; X)$.

While winning strategies are defined in the context of the set X, note that it does not really matter what stages player$_\exists$ plays at, so long as she plays infinitely often. For any property $\mathcal{P}$, a winning strategy for one game converts to a winning strategy for another.

Proposition 3.2.6. *Let X be infinite and co-infinite. Then player$_\exists$ has a winning strategy for $G(\mathcal{P}; X)$ if and only if she has one for $G(\mathcal{P}; \mathrm{Evens})$.*

Proof. Let f be a winning strategy for $G(\mathcal{P}; X)$ and let x_s enumerate the elements of X in order. We construct a strategy g for $G(\mathcal{P}; \mathrm{Evens})$ as follows. Suppose $\langle p_0, \dots, p_{2s-1} \rangle$ is our play so far at the beginning of stage $2s$. For $m < x_s$, define

$$q_m = \begin{cases} p_{2t} & \text{if } m = x_t \\ p_{2t-1} & \text{if } x_{t-1} < m < x_t \end{cases} \tag{3.3}$$

Then we define

$$g(\langle p_0, \dots, p_{2s-1} \rangle) = f(\langle q_0, \dots, q_{x_s - 1} \rangle) \tag{3.4}$$

Let $\langle p_n \rangle$ be any play in $G(\mathcal{P}; \mathrm{Evens})$ satisfying (3.2) for g. Define the play $\langle q_m \rangle$ by (3.3). By definition of g, $q_s = f(\langle q_0, \dots, q_{s-1})$, so $\overline{\mathfrak{A}}(q_\omega)$ has property $\mathcal{P}$. Since $q_\omega = p_\omega$, g is a winning strategy for $G(\mathcal{P}; \mathrm{Evens})$.

The construction of a winning strategy for $G(\mathcal{P}; X)$ from a strategy for $G(\mathcal{P})$ is analogous. $\square$

For this reason, we will often restrict ourselves to the game $G(\mathcal{P}; \mathrm{Evens})$, which we call the *standard game for $\mathcal{P}$*. We denote this game as $G(\mathcal{P})$. By Proposition 3.2.6, we only need to find winning strategies for the standard game.

Definition 3.2.7. A model theoretic property $\mathcal{P}$ is *N-enforceable*, or simply *enforceable*, if player$_\exists$ has a winning strategy for $G(\mathcal{P})$.

To understand why we want enforceable properties, suppose that $\mathcal{P}_0$ and $\mathcal{P}_1$ are both enforceable. Let X be the set of stages divisible by 4, and Y the set of even stages not divisible by 4. By Proposition 3.2.6, player$_\exists$ can transform the winning strategies for the games $G(\mathcal{P}_i)$ into ones for the games $G(\mathcal{P}_0; X)$, $G(\mathcal{P}_1; Y)$. Then if player$_\exists$ plays the appropriate strategies for the each of the stages in X and Y, this is a winning strategy for the standard games of both properties. So there is a structure with both properties.

This gives us an easy way to prove the existence of certain structures. Consider, for example, Theorem 2.5.2. To prove this theorem, we only need to find a notion of forcing N such that both

(a) $\overline{\mathfrak{A}}$ is an e.c. model of T

(b) $\overline{\mathfrak{A}}$ omits Ψ.

are N-enforceable. We can prove (a) and (b) separately for N; Proposition 3.2.6 ensures that there is a structure with all of these properties.

While we have presented model theoretic forcing in the context of winning strategies, it is related to the type of forcing originally developed by Cohen [9]. To see why this is so, we define an analogue of weak forcing.

Definition 3.2.8. For all $p \in N$ and $\varphi \in \mathcal{L}(W)$, we define $p \Vdash \varphi$ as follows.

(a) For φ atomic, $p \Vdash \varphi$ if and only if $p \vdash \varphi$.

(b) For $\varphi = \psi \wedge \chi$, $p \Vdash \varphi$ if and only if $p \Vdash \psi$ and $p \Vdash \chi$.

(c) For $\varphi = \neg\psi$, $p \Vdash \varphi$ if and only if there is no $q \supseteq p$ such that $q \Vdash \psi$

(d) For $\varphi = \exists x\, \psi(x)$, $p \Vdash \varphi$ if and only if there is some closed term τ of $\mathcal{L}(W)$ such that $p \Vdash \psi(\tau)$.

If $p \Vdash \varphi$, we say that p *strongly forces* φ. If $p \Vdash \neg\neg\varphi$, we say that p *weakly forces* φ, which we write as $p \Vdash^\star \varphi$.

Theorem 3.2.9 (Hodges [25]). *For any notion of forcing N and condition $p \in N$, define the notion of forcing*

$$N_p = \{\, q \in N \mid q \supseteq p \,\} \tag{3.5}$$

For any $\varphi \in \mathcal{L}$, $p \Vdash^\star \varphi$ if and only if $\mathfrak{A} \models \varphi$ is N_p-enforceable.

3.3 Forcing and computable structures

We would like to adapt the game in section 3.2 to build computable structures. This is possible because, when the language $\mathcal{L}$ is computable, the construction in Proposition 3.2.3 is effective provided that the $=$-closed set U is computable.

Proposition 3.3.1 (Adapted from Hodges [25]). *Let $\mathcal{L}$ be a computable language with U a computable, $=$-closed set of atomic sentences in $\mathcal{L}$. Then the canonical model $\mathfrak{A}(U)$ is computably presentable.*

Proof. Let X be the set of closed terms in $\mathcal{L}$. If X is empty, then $\mathfrak{A}(U)$ is the empty structure, and we are done. Otherwise, let τ_n be a (possibly finite) computable enumeration of the elements of X. Our structure $\mathfrak{A}$ will have domain

$$A = \{\, n \in \omega \mid (\tau_k = \tau_n) \notin U \text{ for all } k < n \,\} \tag{3.6}$$

Because U is computable, this is a computable subset of $\mathbb{N}$.

For each constant symbol $c \in \mathcal{L}$, we set

$$c^{\mathfrak{A}} = n \quad \text{where} \quad n \in A \text{ and } (\tau_k = c) \in U \tag{3.7}$$

Note that there is such an n because $c = \tau_k$ for some $k \in \mathbb{N}$. Furthermore, this n is unique. Suppose that $n < m$ and $(\tau_n = c)$, $(\tau_m = c) \in D$. Then by (b) in Definition 3.2.2, $(\tau_n = \tau_m) \in D$, and so $m \notin A$. Hence $c^{\mathfrak{A}}$ is well-defined.

Next, let F be any k-ary function symbol with $n_1, \dots, n_k \in A$. We define

$$F^{\mathfrak{A}}(n_1, \dots, n_k) = m \quad \text{where} \quad m \in A\text{and } (t_m = F(\tau_{n_1}, \dots, \tau_{n_k})) \in U \tag{3.8}$$

Again, this definition is well-defined by the argument given for (3.7).

Finally, let R be any k-ary relational symbol and let $n_1, \dots, n_k \in A$. Then we define

$$R^{\mathfrak{A}}(n_1, \dots, n_k) \quad \Leftrightarrow \quad R(\tau_{n_1}, \dots, \tau_{n_k}) \in U \tag{3.9}$$

This completes our specification of $\mathfrak{A}$.

To see that $\mathfrak{A} \models U$, take any $\xi \in U$ of the form $R(\tau_1, \dots, \tau_k)$. It is clear from (3.6) that there are unique elements $n_1, \dots, n_k \in A$ such that $(\tau_{n_i} = \tau_i) \in U$. Then, as U is $=$-closed, $R(\tau_{n_1}, \dots, \tau_{n_k}) \in U$, and so $\mathfrak{A} \models R(\tau_1, \dots, \tau_k)$. A similar argument holds for all atomic sentences $\xi \in U$ of the form $(\tau_1 = \tau_2)$.

Finally, note that for all $n \in A$, $n = \tau^{\mathfrak{A}}$ for some closed term τ of $\mathcal{L}$. Moreover, $(\tau_n = \tau_m) \notin U$ for all n, $m \in A$ with $n \neq m$. So $\mathfrak{A}$ is the least structure satisfying exactly the atomic sentences in U, and hence is the canonical structure $\mathfrak{A}(U)$. $\square$

This means that we should be able to adapt the game in section 3.2 to build computable structures. The obvious way to do this is to restrict that game to effective plays. That is, for any play $\langle p_s \rangle$ there should be some algorithm which computes p_s from the previous choices $p_0, \dots, p_{s-1}$.

In order for such an algorithm to exist, we have to have some way of effectively enumerating all of the elements of N. This is the motivation for the following definition.

Definition 3.3.2. Let $\mathcal{L}$ be a computable language with W a computable set of constants not in $\mathcal{L}$. An *effective notion of forcing* N for $\mathcal{L}$ over W is a notion of forcing such that all the conditions in N are finite and N is a c.e. subset of $\mathcal{L}(W)^{\leq\omega}$.

We have already pointed out why we need N to be a c.e. set. To see why we want the conditions to be finite, note that in order to tell whether a move p_s is allowed, we need to be able to decide whether $p_s \supseteq p_{s-1}$. While it may be possible to decide $p \subseteq q$ for computably infinite conditions given a suitable coding of N, this would make Definition 3.3.2 substantially more complicated. Therefore we restrict ourselves to finite conditions for the sake of simplicity.

For any infinite c.e. set A and partial computable function $\{e\}$, we can define a partial computable function f on A by fixing an enumeration $\{ a_n \}$ of A and defining $f(a_n) = \{e\}(n)$ for all n. Furthermore, f is total exactly when $\{e\}$ is total. This allows us to formulate an effective analogue of a winning strategy.

Definition 3.3.3. Given a model theoretic property $\mathcal{P}$, an *effective winning strategy for $G_N(\mathcal{P}; X)$* is a computable function $f : N^{<\omega} \to N$ such that, whenever $\langle p_s \rangle$ is a computable play satisfying (3.2), $\langle p_s \rangle$ is winning for $G(\mathcal{P}; X)$.

As in the classical case, we do not want to restrict our choice of effective winning strategies to a particular set X. Fortunately, the construction in Proposition 3.2.6 is effective whenever X and the winning strategies are all effective. Since we will restrict ourselves to computable choices of X, we say that $\mathcal{P}$ is *effectively N-enforceable* if there is an effective winning strategy for the standard game $G_N(\mathcal{P})$.

It is important to note that f need not be a winning strategy for the classical game $G_N(\mathcal{P})$; it is only winning when it is used to construct a computable play. This is because we might want our properties to have effective content. For example, $\mathcal{P}$ might be the property that $\overline{\mathfrak{A}}$ is decidable. It is difficult to enforce this property if player$_\forall$ can choose its moves from some set of arbitrary complexity. However, Theorem 3.3.5 demonstrates that it is enough to restrict ourselves to computable plays when we want to construct computable structures.

Definition 3.3.4. Let N be a notion of forcing for $\mathcal{L}$ over W with Ψ a set of sentences in $\mathcal{L}(W)$. N is *Ψ-complete* if, for any $p \in N$ and $\exists_n$ sentence $\varphi \in \Psi$, either $p \cup \{ \varphi \} \in N$ or $p \cup \{ \neg\varphi \} \in N$.

Theorem 3.3.5 (Effective Model Theoretic Forcing). *Let $\Psi = \{ \phi_n \}_{n\in\omega}$ be a c.e. subset of $\mathcal{L}(W)$, and let N be an effective notion of forcing that is Ψ-complete. For any structure $\mathfrak{A}$, we define the predicate*

$$P_{\mathfrak{A}}(n) \quad \Leftrightarrow \quad \mathfrak{A} \models \phi_n \tag{3.10}$$

Then we can effectively enforce the decidability of $P_{\mathfrak{A}}$.

Proof. Our proof is a generalization of the construction that was first carried out by Henkin [23]. We will construct a computable function $f : N^{\leq\omega} \to N$ such that for any computable play $\langle p_s \rangle$ satisfying (3.2), $p = \bigcup_s p_s$ has the following properties.

(a) For every sentence $\neg\neg\varphi \in p$, $\varphi \in p$.

(b) For every sentence $\varphi \wedge \psi \in p$, both $\varphi \in p$ and $\psi \in p$.

(c) For every sentence $\neg(\varphi \wedge \psi) \in p$, either $\neg\varphi \in p$ or $\neg\psi \in p$.

(d) For each sentence $\exists x\, \varphi(x) \in p$, there is some closed term τ in $\mathcal{L}(W)$ such that $\varphi(t) \in p$.

(e) For each sentence $\neg\exists x\, \varphi(x) \in p$ and all closed terms τ in $\mathcal{L}(W)$, $\neg\varphi(\tau) \in p$.

(f) For all $\varphi \in \Psi$, either $\varphi \in p$ or $\neg\varphi \in p$.

In this case, $P_{\mathfrak{A}}$ will be decidable for the compiled structure $\overline{\mathfrak{A}}(p)$.

To see why this is so, we first show that $\overline{\mathfrak{A}}(p) \vDash p$. We proceed by induction on the complexity of $\varphi \in p$. If φ is atomic, there is nothing to prove. The properties (b) and (d) take care of the cases $\varphi = \psi \wedge \chi$ and $\varphi = \exists x\, \psi(x)$, respectively. So the only nontrivial case is when $\varphi = \neg\psi$. There are four possibilities for ψ.

Case ψ is atomic: As $\neg\psi \in p$, $\psi \notin p^{\mathrm{atm}}$. Since $\mathfrak{A}(p)$ is the canonical model of p^{atm}, it satisfies exactly the atomic sentences in this set. So $\overline{\mathfrak{A}}(p) \nvDash \psi$ and hence $\overline{\mathfrak{A}}(p) \vDash \neg\psi$.

Case $\psi = \neg\chi$: As $\neg\psi \in p$, $\neg\neg\chi \in p$. So by property (a), $\chi \in p$. Then $\overline{\mathfrak{A}}(p) \vDash \chi$ by induction, and so $\overline{\mathfrak{A}}(p) \vDash \neg\psi$.

Case $\psi = \chi \wedge \rho$: As $\neg\psi \in p$, $\neg(\chi \wedge \rho) \in p$. So by property (c), either $\neg\chi \in p$ or $\neg\rho \in p$. Assume without loss of generality that $\neg\chi \in p$. Then $\overline{\mathfrak{A}}(p) \vDash \neg\chi$ by induction, and so $\overline{\mathfrak{A}}(p) \vDash \neg\psi$.

Case $\psi = \neg\exists x\, \chi(x)$: Suppose for a contradiction that $\overline{\mathfrak{A}}(p) \vDash \neg\psi$. Then $\overline{\mathfrak{A}}(p) \vDash \exists x\, \chi(x)$. We know from Proposition 3.2.3 that every element $a \in \mathfrak{A}$ has the form $\tau^{\overline{\mathfrak{A}}(p)}$ for some closed term τ of $\mathcal{L}(W)$. So there is some closed term τ such that $\overline{\mathfrak{A}}(p) \vDash \chi(\tau)$. However, by property (e), $\neg\chi(\tau) \in p$. This contradicts our inductive hypothesis.

As $\overline{\mathfrak{A}}(p) \vDash p$, the decidability of $P_{\overline{\mathfrak{A}}(p)}$ follows from (f) and the fact that $p = \bigcup_s p_s$ is c.e. So we need only construct our computable function f satisfying the above properties.

Since f is defined on sequences in N, instead of just elements of N, we can incorporate the stage of the play into our strategy. We will break the even stages up into several groups, and assign each group the task of ensuring one of the properties (a) – (f). Define the set

$$\Upsilon = \{\, \psi \in p \mid p \in N \,\} \tag{3.11}$$

Since N is computable and all of the conditions in N are finite, Υ is a computable set. Let ψ_n be a computable enumeration of the elements of Υ.

Stage $s = 2(6e)$:
At this stage, we ensure that p is complete with respect to Ψ. As above, let θ_e be the e^{th} element of Ψ. Since $p_{s-1} \in N$ and N is Ψ-complete, either $p_{s-1} \cup \{\, \phi_e \,\} \in N$ or $p_{s-1} \cup \{\, \neg\phi_e \,\} \in N$. We extend p_s to the least such condition in N.

Stage $s = 2(6\langle n, k\rangle + 1)$:

At this stage, we ensure that p has property (a). As above, let ψ_n be the k^{th} element of Υ. If ψ_k is not of the form $\neg\neg\chi$, then there is nothing to do. In this case we just set $p_s = p_{s-1}$.

Otherwise, we check whether $\psi_k \in p_{s-1}$. If not, then again there is nothing to do. So suppose $\psi_k \in p_{s-1}$. Since $p_{s-1} \in N$ and N is a notion of consistency, we have that $p_{s-1} \cup \{ \chi \} \in N$. We extend p_s accordingly.

Stage $s = 2(6\langle n, k\rangle + 2)$:

At this stage, we ensure that p has property (b). As with stage, $2(6\langle n, k\rangle + 1)$, there is nothing to do unless $\psi_k = \chi \wedge \rho$ and $\psi_k \in p_{s-1}$. In this case, since $p_{s-1} \in N$ and N is a notion of consistency, we have that $p_{s-1} \cup \{ \chi, \rho \} \in N$. We extend p_s accordingly.

Stage $s = 2(6\langle n, k\rangle + 3)$:

At this stage, we ensure that p has property (c). As with stage, $2(6\langle n, k\rangle + 1)$, there is nothing to do unless $\psi_k = \neg(\chi \wedge \rho)$ and $\psi_k \in p_{s-1}$. In this case, we extend p_s to either $p_{s-1} \cup \{ \neg\chi \} \in N$ or $p_{s-1} \cup \{ \neg\rho \} \in N$.

Stage $s = 2(6\langle n, k\rangle + 4)$:

At this stage, we ensure that p has property (d). As with stage, $2(6\langle n, k\rangle + 1)$, there is nothing to do unless $\psi_k = \exists x\, \chi(x)$ and $\psi_k \in p_{s-1}$. In this case, there is some closed term τ of $\mathcal{L}(W)$ such that $p_{s-1} \cup \{ \chi(\tau) \} \in N$. We extend p_s accordingly.

Stage $s = 2(6\langle n, \langle k, \ell\rangle\rangle + 5)$:

At this stage, we ensure that p has property (e). As with stage, $2(6\langle n, k\rangle + 1)$, there is nothing to do unless $\psi_k = \neg\exists x\, \chi(x)$ and $\psi_k \in p_{s-1}$. Let τ_ℓ be the ℓ^{th} closed term of $\mathcal{L}(W)$. We extend p_s to $p_{s-1} \cup \{ \neg\chi(\tau_\ell) \} \in N$.

This procedure defines a computable function $f : N^{\leq\omega} \to N$. To see that f is an effective winning strategy, let $\langle p_s\rangle$ be a computable play statisfying (3.2). Clearly the stages $2(6e)$ ensure that $p = \bigcup_s p_s$ is complete with respect to Ψ. So we need only show that p has properties (a) – (e).

We will show property (e); the other proofs are all analogous. Suppose that $\neg\exists x\, \varphi(x) \in p$. There is some stage s such that $\neg\exists x\, \varphi(x) \in p_s$. Let n_0 be such that $s \leq 2(6\langle n_0, \langle k, \ell\rangle\rangle + 5)$ for all k, $\ell \in \omega$. Note that $\neg\exists x\, \varphi(x) = \psi_{k_0}$ for some $k_0 \in \omega$. Then, for each closed term τ_ℓ, we ensure $\neg\varphi(\tau_\ell) \in p$ at stage $2(6\langle n_0, \langle k_0, \ell\rangle\rangle+5)$. $\square$

Corollary 3.3.6. *Let N be an effective notion of forcing that is complete for atomic sentences, and let $\mathcal{P}$ be an effectively enforceable model theoretic property. Then there is a computable structure $\mathfrak{A}$ with property $\mathcal{P}$.*

Proof. Since the set of atomic sentences is computable, we know from Theorem 3.3.5 that we can effectively enforce the decidability of $P_{\mathfrak{A}}$ for atomic sentences. Let f be the effective winning strategy for the decidability of $P_{\mathfrak{A}}$, and let g be the effective winning strategy for $\mathcal{P}$. We will use these two computable functions to construct a computable winning play $\langle p_s\rangle$ for $G(\mathcal{P})$ such that the compiled structure $\overline{\mathfrak{A}}(p_\omega)$ is computably presentable.

By Proposition 3.2.6, we can assume without loss of generality that g is actually an effective winning strategy for the game $G(\mathcal{P}, \mathrm{Odds})$. We construct a play where player$_\forall$ follows strategy g, while player$_\exists$ follows strategy f. That is, for all $s \in \omega$,

$$p_s = \begin{cases} g(\langle p_0, \dots, p_{s-1}\rangle) & \text{if } s \text{ odd} \\ f(\langle p_0, \dots, p_{s-1}\rangle) & \text{otherwise} \end{cases} \tag{3.12}$$

Since f, g are both computable, $\langle p_s \rangle$ is a computable play. Hence, by definition, this is a winning play for $G(\mathcal{P})$ and so $\mathfrak{A}(p_\omega)$ has property $\mathcal{P}$. $\mathfrak{A}(p_\omega)$ is computable, since f ensures that p_ω is complete for atomic sentences. $\square$

This theorem and its accompanying corollary are particularly powerful results, and we will use them to construct all sorts of computable structures. In particular, Theorem 3.1.1, the effective completeness theorem, is a straightforward corollary of this theorem. If T is a decidable theory, then the notion of forcing defined N_T^{con} defined in (3.1) is effective and complete for all elements of $\mathcal{L}(W)$.

It is important to point out that completeness for atomic sentences is necessary for Corollary 3.3.6. For any given computable play $\langle p_s \rangle$, p_ω is c.e., but not necessarily computable. Consider a language $\mathcal{L}$ with an infinite number of unary relations $\{ P_n \}$. Let $\mathcal{L}^{\mathrm{atm}}(W)$ be the set of all atomic sentences in $\mathcal{L}(W)$. It is easy to check that the set

$$N = \left\{ p \subset \mathcal{L}^{\mathrm{atm}}(W) \mid p \text{ is finite and } P_n(c) \in p \Rightarrow n \in A \right\} \tag{3.13}$$

is an effective notion of forcing. However, player$_\forall$ can always guarantee that

$$p_\omega = \{ P_n(c) \mid n \in A \text{ and } c \in W \} \tag{3.14}$$

by adding the missing atomic sentences at the odd stages. The canonical model of this set is not computably presentable, as we can recover A from it.

As in the classical case, we want to combine effectively enforceable properties in our computable structures. When we are presented with a finite list $\mathcal{P}_0, \dots, \mathcal{P}_n$ of effectively enforceable properties, this is not a problem. We break the even stages up into a partition of n infinite sets and use Proposition 3.2.6 to transform the winning strategies for each of the games $G(\mathcal{P}_k)$, appropriately.

However, sometimes, such as in Vaught's [48] Omitting Types Theorem, we want to combine an infinite number of enforceable properties. If we try to do this effectively, we run into problems with uniformity. In this case, we need to know something about the complexity of the set of properties itself, as demonstrated in the effective analogue of the Omitting Types Theorem.

Theorem 3.3.7 (Millar [35]). *Let T be a complete decidable theory. If Φ_1 is a Σ^0_2 set of computable nonprincipal types of T, and Φ_2 is a Σ^0_2 set of complete types of T, then there is a decidable model of T which omits all types in Φ and all nonprincipal types in Ψ.*

Recall that a Σ_2^0 set of types is a Σ_2^0 set $X \subseteq \omega$ such that for every $e \in X$, $\{e\}$ is total and codes the characteristic function of a type over T. This theorem works because we can associate each $e \in X$ with a set of formulae over $\mathcal{L}$. We approximate X and attempt to omit the set of formulae represented by $\{e\}$ whenever we suspect that $e \in X$. This idea is the motivation for the following definition.

Definition 3.3.8. Let $\{\mathcal{P}_n\}$ be a countable list of model theoretic properties with $E \subseteq \omega$. A *uniformly effective winning strategy over* E is a partial computable function on $N \times N^{<\omega}$ into N such that for all $n \in E$, f_n is an effective winning strategy for $\mathcal{P}_n$. If such an f exists, we say that the properties $\{\mathcal{P}_n\}$ are *uniformly enforceable over* E.

Given this definition, we now have a way to uniformly combine an infinite set of effectively enforceable properties. The following theorem will be very useful when we look at several effective omitting types constructions in Chapter 4.

Theorem 3.3.9. *Let* $\{\mathcal{P}_k\}$ *be a countable list of model theoretic properties with a* Σ_2^0 *set* $E \subseteq \omega$ *such these properties are uniformly enforceable over* E. *Then we can effectively enforce a computable structure* $\mathfrak{A}$ *with all of the properties* $\mathcal{P}_n$ *for* $n \in E$.

Proof. Our proof is a generalization of the construction in Theorem 3.3.7. Let f be the partial computable function on $N \times N^{<\omega}$ given by Definition 3.3.8. Also let $R(x, y, n)$ be a computable predicate such that $n \in E \Leftrightarrow \exists x \, \forall y \, R(x, y, n)$. We construct a new effective winning strategy g by specifying our actions at various even stages, as we did in Theorem 3.3.5.

Stage $s = -1$:

When $n \in E$, the function f_n is an effective winning strategy for $G(\mathcal{P}_n)$. As in Corollary 3.3.6, in order to combine these strategies, we will need to effectively transform them into effective winning strategies of games $G(\mathcal{P}_n; X_n)$. These sets X_n will change as we approximate E, so we define a c.e. collection of computable infinite and coinfinite sets X_n^s. Initially, we define

$$X_n^{-1} = \{ \langle n, k \rangle \mid k \in \omega \} \tag{3.15}$$

for all $n \in \omega$.

We also need a function $x_s(n)$ which will be used to approximate $n \in E$. The idea is that, if $n \in E$, then eventually there will be some stage s at which $\forall y \, R(x_s(n), y, n)$ holds. Initially, we set $x_{-1}(n) = 0$ for all n.

Stage $s = \langle n, k \rangle$:

At this stage we try to ensure that property $\mathcal{P}_n$ is met. If we consider $\langle p_0, \ldots, p_{s-1} \rangle$ as a sequence in the game $G(\mathcal{P}_n; X_n^s)$, we use the techniques of Proposition 3.2.6 to convert it to an equivalent sequence $\langle q_0, \ldots, q_{r-1} \rangle$ in the game $G(\mathcal{P}_n)$. As f is partially computable, we simultaneously search for a $q \in N$ such that

$$q = f_n(\langle q_0, \ldots, q_{r-1} \rangle) \tag{3.16}$$

and a $y \in \mathbb{N}$ such that $R(x_s(n), y, n)$ fails to hold.

Eventually one of these two searches will halt. If there is no q satisfying (3.16), then f_n is not total and so it cannot be an effective winning strategy for $G(\mathcal{P}_n)$. In this case $n \notin E$ and so $\forall x\, \exists y\, \neg R(x, y, n)$. Then there is some y such that $R(x_s(n), y, n)$ fails to hold.

If we first find some $q \in N$ satisfying (3.16), set $p_s = q$ and leave x_s, X_n^s all unchanged. Otherwise, we leave p_s unchanged. We also increment $x_s(n)$ and set

$$X_n^{-1} = \{\, \langle n, k\rangle \mid \langle n, k\rangle \geq s \,\} \tag{3.17}$$

We leave $x_s(m)$ and X_m^s unchanged for all other values of m.

Let $\langle p_s\rangle$ be any play satisfying (3.2) for the strategy defined above, and suppose that $n \in E$. Then eventually there is some stage s such $\forall y\, R(x_t(n), y, n)$ holds for all $t > s$. So $X_n = X_n^s$ is never again reset after that stage. Futhermore, at all stages $t > s$ in X_n, we are applying an effective winning strategy for the game $G(\mathcal{P}_n; X_n)$. So the compiled structure $\overline{\mathfrak{A}}(p_\omega)$ has property $\mathcal{P}_n$. $\square$

Chapter 4

Computable Models of $\forall_2$ Theories

4.1 Introduction

In this chapter, we will use the techniques of effective model theoretic forcing to construct computable models of $\forall_2$ theories. Our goal is to establish a characterization that is an analogue of Theorem 3.1.1 for theories with computable models. To some degree, this has already been done in the following result of Baur [6].

Definition 4.1.1. Let $\{\phi_k\}_{k\in\omega}$ be an enumeration of the $\exists_n(\forall_n)$ sentences of $\mathcal{L}$. A theory T is $\exists_n(\forall_n)$*-c.e.* if the predicate $\mathcal{P}(k) =_{\text{def}} (T \vdash \phi_k)$ is c.e. Similarly, T is $\exists_n(\forall_n)$*-decidable* if $\mathcal{P}$ is decidable.

Theorem 4.1.2 (Baur [6]). *Let T be a theory with a Δ^0_2 list of $\forall_2$ axioms. Then T has a computable model if and only if T has a consistent 1-completion $\widehat{T}$ such that $\widehat{T}$ is $\exists$-c.e.*

However, there are some features of this characterization that keep it from being a satisfying analogue of Theorem 3.1.1. In the effective completeness theorem, there are no effective conditions on the theories to which it applies. It is a characterization for all complete theories, regardless of their algorithmic complexity. The above theorem, however, is only a characterization for $\forall_2$ theories with a Δ^0_2 list of axioms. As we see in the following example, there are $\forall_2$ theories with computable models which do not satisfy this hypothesis.

Example 4.1.3. There is a $\forall_2$ theory T with a computable model $\mathfrak{A}$ such that there is no consistent $\forall_2$ extension $\widehat{T} \supseteq T$ with a Δ^0_2 list of axioms.

Proof. Let $\mathbb{N}$ be the standard model of arithmetic with $+$, $\cdot$, 0, and 1. Let $T = \text{Th}_{\forall_2}(\mathbb{N})$. Clearly T has a computable model. Furthermore, T does not have a Δ^0_2 list of axioms. If it did, then set of the consequences of T would be c.e. in this list of axioms, and hence Σ^0_2. However, the set of $\forall_2$ sentences true in $\mathbb{N}$ is Π^0_2-complete.

We need only show that T is 2-complete; if this is true, then there is no proper, consistent $\forall_2$ extension $\widehat{T} \supseteq T$. Take any $\forall_2$ sentence $\varphi = \forall \overline{x}\, \psi(\overline{x})$, where ψ is an $\exists$ formula. We need to show that either $T \vdash \varphi$, or $T \vdash \neg\varphi$.

If $\mathbb{N} \models \varphi$, then clearly $T \vdash \varphi$. So suppose $\mathbb{N} \not\models \varphi$. Then there is some $\overline{a} \in \mathbb{N}$ such that $\mathbb{N} \models \neg\psi(\overline{a})$. As all of the elements of $\overline{a}$ are representable by closed terms in this language, $T \vdash \neg\psi(\overline{a})$, and so $T \vdash \neg\varphi$. □

It is still an open question whether or not the characterization in Theorem 4.1.2 can be extended to all $\forall_2$ theories. In particular, it is unknown whether there is a theory T with an $\exists$-c.e. 1-completion but no computable model.

Another problem with this result is that it only characterizes computable models. However, as we pointed out in Chapter 2, when we study $\forall_2$ theories, we are primarily interested in e.c. models. Unfortunately, as our next example shows, Theorem 4.1.2 does not characterize the existence of computable e.c. models.

Example 4.1.4. There is a c.e. $\forall_2$ theory with a computable model, but no computable existentially closed model.

Proof. Our language will consist of three binary relations, R, Q, and E. For each $n \in \omega$, let $\varphi_n^R(x)$ say that x is an element of an R-cycle of length n. That is,

$$\varphi_n^R(x) =_{\text{def}} \exists y_1, \ldots y_n \big(\bigwedge_{i \leq n} y_i \; R \; y_{i+1} \wedge y_1 = x \wedge y_n = x \big) \tag{4.1}$$

Similarly, define $\varphi_n^Q(x)$. Let X and Y be a pair of computably inseparable c.e. sets. Our theory T is given by the following list of axioms.

(a) R, Q,and E are all irreflexive. That is, $\forall x\, (\neg x \; R \; x)$, and similarly for both Q and E.

(b) Every element has at most one R predecessor and at most one R successor. That is,

$$\forall x, y, z \big(((y \; R \; x \wedge z \; R \; x) \to y = z) \wedge ((x \; R \; y \wedge x \; R \; z) \to y = z) \big) \tag{4.2}$$

(c) Every element has at most one Q predecessor and at most one Q successor.

(d) Every point with an E successor is E-adjacent to some R-cycle of length n, for $n \in X$. That is,

$$\forall x \big(\exists y\, x \; E \; y \to \exists z\, (x \; E \; z \wedge \varphi_n^R(z)) \big) \tag{4.3}$$

for all $n \in X$.

(e) Every point with an E successor is E-adjacent to some Q-cycle of length n, for $n \in Y$.

(f) No element is E-adjacent to both an R-cycle of length n and a Q-cycle of length n. That is,

$$\forall x \left(\forall y\,(x \; E \; y \to \neg\varphi_n^R(y)) \wedge \forall z\,(x \; E \; z \to \neg\varphi_n^Q(z))\right) \tag{4.4}$$

This theory is clearly $\forall_2$, and it is c.e. since both X and Y are. It also has a computable model, namely the trivial model with one element c. However, it has no computable e.c. model.

Suppose for a contradiction that $\mathfrak{A}$ is a computable e.c. model of T. Clearly $\exists x, y(x \; E \; y)$ is consistent with T; we just construct a model with a basepoint that is E-connected to an R-cycle of length n for each $n \in X$ and E-connected to a Q-cycle of length k for each $k \notin X$.

As $\mathfrak{A}$ is e.c., $\mathfrak{A} \vDash \exists x, y(x \; E \; y)$. Let $a \in \mathfrak{A}$ be some element with an E-successor. For all $n \in \omega$, a must either be E-connected to either an R-cycle or a Q-cycle of length n. If it is not connected to either, then we can extend $\mathfrak{A}$ to a new model $\mathfrak{B} \vDash T$ with $\mathfrak{B} \vDash \exists z(a \; E \; z \wedge \varphi_n^R(z))$ by adding a new R-cycle to $\mathfrak{A}$ and attaching it to a. This contradicts the fact the $\mathfrak{A}$ is existentially closed.

Let Z be the set of all n such that a is E-connected to an R-cycle of length n. Z is computable. For each n, we search until we find either a R-cycle or a Q-cycle of length n E-connected to a. If we find a Q-cycle, there can be no R-cycle, and hence $n \notin Z$. However, Z is a separator of X and Y, which is a contradiction. $\square$

For the rest of this chapter, we will attempt to address these inadequacies of Theorem 4.1.2. We will prove several analogues of the effective completeness theorem for different types of theories and computable models. Afterwards, we will use these characterizations to construct computable models that have additional properties, such as being elementarily prime or omitting a type.

4.2 Forcing computable e.c. models

Our first goal is to find some sufficient criteria for the existence of a computable e.c. model. To do this, we look to classical constructions of e.c. models. Generally, e.c. models are constructed via a specific type of model theoretic forcing called Robinson forcing.

Definition 4.2.1. Let T be a theory with W and infinite set of constants not appearing in T. The *notion of Robinson forcing* for T is the set

$$N_T = \{\, p \subseteq \mathcal{L}(W) \mid p \text{ is finite set of literals and } T \cup p \text{ is consistent.} \,\} \tag{4.5}$$

It is clear that N_T is a notion of forcing as defined in Definition 3.2.4.

The idea is that, for any model $\mathfrak{A} \vDash T$, its atomic diagram is the union of some sequence of elements $p_s \in N_T$. These sequences may also give us other structures that are not models of T, but we can eliminate those with the appropriate strategy.

Let $\overline{\mathfrak{A}}$ be the compiled structure of P_ω as in Chapter 3. To ensure that $\overline{\mathfrak{A}}$ is an e.c. model of T, we need only enforce, for each $\overline{c} \in W$ and existential formula $\varphi(\overline{x}) \in \mathcal{L}$, the property

$$\exists \mathfrak{B} \supseteq \overline{\mathfrak{A}} \left(\mathfrak{B} \models T \wedge \mathfrak{B} \models \varphi(\overline{c}) \right) \quad \Rightarrow \quad \overline{\mathfrak{A}} \models \varphi(\overline{a}) \tag{4.6}$$

Since every model of T is given by some sequence of $p_s \in N_T$, there is an easy winning strategy for this game. Given $p \in N_T$, we check whether there is some extension $q \supseteq p$ such that $\overline{\mathfrak{A}}(q) \models \varphi(\overline{c})$. If there is no such q, then there is no $\mathfrak{B} \models T \cup p$ with $\mathfrak{B} \models \varphi(\overline{c})$ and hence (4.6) is vacuously satisfied. Otherwise, choosing $q \supseteq p$ ensures that $\overline{\mathfrak{A}} \models \varphi(\overline{c})$.

If we want to build computable e.c. models, then we just make this strategy effective, and use the results of section 3.3. To do this, we first need N_T to be an effective notion of forcing. This requires some effective content in the theory T. Assume from now on that our language $\mathcal{L}$ is computable.

Proposition 4.2.2. *Let T be a consistent theory. N_T is computable if and only if T is $\forall$-decidable.*

Proof. ($\Rightarrow$) Take any $\forall$ sentence φ. We use prenex and disjunctive normal form to rewrite $\neg\varphi$ as

$$\neg\varphi \equiv \exists \overline{x} \bigvee_{i}^{n} \bigwedge_{j}^{m_i} \psi_{i,j}(\overline{x}) \tag{4.7}$$

where the $\psi_{i,j}$ are all literals. Let W be the infinite set of constants from Definition 4.2.1, and take any $\overline{c} \in W$. Then

$$T \cup \{ \neg\varphi \} \text{ is consistent} \quad \Leftrightarrow \quad \{ \psi_{i,j}(\overline{c}) \}_{j \le m_i} \in N_T \text{ for some } i \le n \tag{4.8}$$

As $T \vdash \varphi$ if and only if $T \cup \{ \neg\varphi \}$ is not consistent, we are done.

($\Leftarrow$) Take any finite set of literals p with $\delta(\overline{c}) = \wedge p$. Then

$$p \in N_T \quad \Leftrightarrow \quad T \nvdash \forall \overline{x}\, \neg\delta(\overline{x}) \tag{4.9}$$

□

So N_T is an effective notion of forcing whenever T is $\forall$-decidable. Therefore, for the rest of this section we will be examining $\forall$-decidable $\forall_2$ theories.

It is important to note that $\forall$-decidability is a very different notion from the decidability of $T_\forall$. For example, suppose T is the theory generated by the empty set over the language $\mathcal{L}$ with a single binary relation R. Then $T = T_\forall$, and we know that T is not necessarily decidable, since first order logic is not decidable for this language. However, T is $\forall_2$-decidable by the following result.

Theorem 4.2.3 (Bernays and Schönfinkel [7]). *There is an algorithm which is uniform in $\mathcal{L}$ for deciding whether φ has a model for each $\exists_2$ sentence $\varphi \in \mathcal{L}$.*

Another important observation is that we do not need $\forall$-decidability to make N_T an effective notion of forcing. Recall from Definition 3.3.2 that we only need for N_T to be c.e., not computable. Following the proof of Proposition 4.2.2, we see that N_T is c.e. whenever the predicate $T \vdash \phi_k$ is co-c.e. for universal ϕ_k. However, as we shall see later, this notion is too weak to make our winning strategies effective; therefore we will not explore this point any further.

Most of our strategies will consist of providing witnesses for consistent $\exists$ sentences. In order to make this effective, we need the following result.

Proposition 4.2.4. *Let T be a consistent $\forall$-decidable theory. Given any $p \in N_T$ and $\exists$ sentence $\varphi \in \mathcal{L}(W)$ consistent with $T \cup p$, we can effectively find some $q \supseteq p$ in N_T such that $q \vdash \varphi$.*

Proof. Suppose we are given $p \in N_T$ and $\exists$ sentence $\varphi \in \mathcal{L}(W)$ with $\varphi = \exists \overline{x}\, \psi(\overline{x})$. Since p is finite, there is some sequence of constants $\overline{c} \in W$ not appearing in p or T. For any such $\overline{c} \in W$

$$\begin{aligned} T \cup p \cup \{\, \varphi \,\} \text{ is consistent} &\Leftrightarrow T \cup p \nvdash \forall \overline{x}\, \neg\psi(\overline{x}) \Leftrightarrow T \cup p \nvdash \neg\psi(\overline{c}) \\ &\Leftrightarrow T \cup p \cup \{\, \psi(\overline{c}) \,\} \text{ is consistent} \end{aligned} \tag{4.10}$$

Again, we again use prenex and disjunctive normal form to rewrite $\psi(\overline{c})$ as

$$\psi(\overline{c}) \equiv \bigvee_{i}^{n} \bigwedge_{j}^{m_i} \psi_{i,j}(\overline{c}) \tag{4.11}$$

As $T \cup p \cup \{\, \psi(\overline{c}) \,\}$ is consistent, there is some $i \leq n$ such that

$$p \cup \{\, \psi_{i,j}(\overline{c}) \,\}_{j \leq m_i} \in N_T \tag{4.12}$$

As N_T is computable, we search for the least such i and extend q accordingly. ☐

With this, we can now make the strategy that we outlined above effective.

Theorem 4.2.5. *Let T be a consistent, $\forall$-decidable $\forall_2$ theory. Then we can effectively enforce an existentially closed model of T.*

Proof. Given the equivalent definitions of existentially closed models in Theorem 2.2.3, it is enough to show that the following two properties are effectively enforceable over N_T.

(a) $\overline{\mathfrak{A}} \vDash T_\forall$.

(b) Let $\varphi(\overline{x})$ be a universal formula. If $\overline{\mathfrak{A}} \vDash \varphi(\overline{a})$ for some $\overline{a} \in \overline{\mathfrak{A}}$, then there is an $\exists$ formula $\psi(\overline{x})$ with $\overline{\mathfrak{A}} \vDash \psi(\overline{a})$ such that

$$T \vdash \forall \overline{x}\, (\psi(\overline{x}) \to \varphi(\overline{x})) \tag{4.13}$$

To see that property (a) is effectively enforceable, note that we can effectively enforce that p_ω is complete for atomic sentences. Let θ_n be an effective enumeration of the atomic sentences of $\mathcal{L}(W)$. At each even stage $2s$, we check whether $\theta_s \in p_{2s-1}$. If not, then either $p_{2s-1} \cup \{ \theta_s \} \in N_T$ or $p_{2s-1} \cup \{ \neg\theta_s \} \in N_T$. We enumerate N_T until we find the least one and extend p_{2s-1} accordingly.

Now let $\overline{\mathfrak{A}}$ be the compiled structure of a winning play $\langle p_s \rangle$. By the construction in Proposition 3.2.3, $\operatorname{diag}(\overline{\mathfrak{A}}) = p_\omega$. By compactness, $T \cup p_\omega$ is consistent, so there is some model $\mathfrak{B} \models T \cup \operatorname{diag}(\overline{\mathfrak{A}})$. As $\overline{\mathfrak{A}}$ is the substructure of a model of T, $\overline{\mathfrak{A}} \models T_\forall$.

To construct an effective winning strategy for property (b), it is enough to construct one for a fixed universal formula $\varphi(\overline{x})$ of $\mathcal{L}(W)$ and a fixed sequence of constants $\overline{a} \in W$; we can use Theorem 3.3.9 to put these strategies together for all universal formula and constants in W. Suppose we are given $p \in N_T$. Set $\delta_p(\overline{a}, \overline{b}) = \wedge p$. We use the $\forall$-decidability of T to determine whether

$$T \vdash \forall \overline{x} \left(\delta_p(\overline{a}, \overline{x}) \to \varphi(\overline{a}) \right) \tag{4.14}$$

If so, we are done. Otherwise, by Proposition 4.2.4 we can effectively find some $q \supseteq p$ which makes $\neg\phi(\overline{a})$ true. We extend p accordingly. $\square$

This proof demonstrates why we require that T be $\forall$-decidable. In order to effectively enforce property (b), we need to take action whenever the predicate in (4.14) fails. As we pointed out earlier, if N_T is c.e. but not computable, then the set of universal consequences of T is properly co-c.e., and hence we cannot delay our move until learn that (4.14) fails. Furthermore, if we play on and try to take action at a later stage, then δ_p will have changed and we will need to decide a new instance of (4.14). $\forall$-decidability allows us to take action immediately, circumventing this problem.

$\forall$-decidability is a fairly strong requirement. The theory in Example 4.1.4 is c.e., but it is not $\forall$-decidable since it does not have a computable e.c. model. In order for Theorem 4.2.5 to be useful, we need to know that there are $\forall$-decidable $\forall_2$ theories. Fortunately, such theories do occur naturally. In particular, the theory T_{acf} of algebraically closed fields is both $\forall_2$ and decidable (See Heintz [22] and Chistov and Grigor'ev [8] for two possible decision procedures). Furthermore, the theory T_{f} of fields is also $\forall_2$, and we know from a result of Robinson [41] that $(T_{\text{acf}})_\forall = (T_{\text{f}})_\forall$. So T_{f} is also $\forall$-decidable.

$\forall$-decidability is also interesting because, as the following example shows, there are $\forall$-decidable $\forall_2$ theories with no decidable extension. This means that Theorem 4.2.5 provides sufficient criteria for the existence of a computable model that are not given by Theorem 3.1.1.

Example 4.2.6. There is a consistent, $\forall$-decidable $\forall_2$ theory T with no c.e. extension.

Proof. Our language will consist of infinitely many binary relations, $Q_{n,k}$ and R_n for all n, $k \in \omega$. Let X be a Π^0_1 subset of ω that is not computable. As $\overline{X}$ is c.e., we can choose some uniformly computable sequence of disjoint, finite sets Y_s such that $\overline{X} = \bigcup_s Y_s$. We will use the sets Y_s to construct a consistent, c.e., 1-complete set of sentences. This set will form the foundation for the theory which we will define later.

For each n, $k \in \omega$, define the set

(4.15)
$$\Psi_{n,k} = \begin{cases} \{\, \exists x, y\, (x\ Q_{n,k}\ y),\ \forall x\, \big(\exists y\, (x\ Q_{n,k}\ y) \to \forall z\, (\neg x\ R_n\ z)\big) \,\} & \text{if } n \in Y_k \\ \{\, \exists x, y\, (x\ Q_{n,k}\ y) \,\} & \text{otherwise} \end{cases}$$

From these finite sets of sentences, we define the set of sentences $\Phi = \bigcup_{\langle n,k\rangle} \Psi_{n,k}$. Finally, we define the set

$$\widehat{\Phi} = \Phi \cup \{\, \varphi \mid \varphi \text{ is existential and } \Phi \cup \{\, \varphi \,\} \text{ is consistent } \} \tag{4.16}$$

We need to show that this set has the desired properties.

Lemma 4.2.7. *$\widehat{\Phi}$ is a consistent, 1-complete set of sentences.*

Proof. This set is clearly 1-complete. We need only show that it is consistent. Clearly Φ is consistent; it has a trivial model $\mathfrak{A}$ of one element c where $(c\ Q_{n,k}\ c)$ and $(\neg c\ R_n\ c)$. So, for each existential sentence φ, we let $\mathfrak{A}_\varphi$ be a model of Φ such that $\mathfrak{A}_\varphi \vDash \varphi$ whenever $\Phi \cup \{\, \varphi \,\}$ is consistent.

Let $\mathfrak{B} = \bigsqcup_\varphi \mathfrak{A}_\varphi$. In other words, $\mathfrak{B}$ is the unique structure such that

(a) The domain of $\mathfrak{B}$ is the disjoint union of the domains of the $\mathfrak{A}_\varphi$.

(b) For all atomic formulae $\xi(\overline{x})$ and elements $\overline{b} \in \mathfrak{B}$, $\mathfrak{B} \vDash \xi(\overline{b})$ if and only if there is some φ such that $\overline{b} \in \mathfrak{A}_\varphi$ and $\mathfrak{A}_\varphi \vDash \xi(\overline{b})$.

Note then that $\mathfrak{A}_\varphi \subseteq \mathfrak{B}$ for all φ. As existential sentences are preserved upwards, this means that $\mathfrak{B} \vDash \varphi$ for all $\exists$ sentences $\varphi \in \widehat{\Phi}$. So we need only show that

$$\mathfrak{B} \vDash \forall x\, \big(\exists y\, (x\ Q_{n,k}\ y) \to \forall z\, (\neg x\ R_n\ z)\big) \tag{4.17}$$

for all $n \in Y_k$.

Take any n, k such that $n \in Y_k$, and any $b \in \mathfrak{B}$ such that $\mathfrak{B} \vDash \exists y\, (b\ Q_{n,k}\ y)$. Let $a \in \mathfrak{B}$ be such that $\mathfrak{B} \vDash (b\ Q_{n,k}\ a)$. Let φ be such that $b \in \mathfrak{A}_\varphi$. Then, by definition, $a \in \mathfrak{A}_\varphi$. Now take any $c \in \mathfrak{B}$. If $c \notin \mathfrak{A}_\varphi$ then $\mathfrak{B} \vDash (\neg b\ R_n\ c)$ by definition. Otherwise, since $\mathfrak{A}_\varphi \vDash \exists y\, (b\ Q_{n,k}\ y)$ and $\mathfrak{A}_\varphi \vDash \Phi$, $\mathfrak{A}_\varphi \vDash (\neg b\ R_n\ c)$. So $\mathfrak{B} \vDash \forall z\, (\neg b\ R_n\ z)$, and hence (4.17) holds. ■

Lemma 4.2.8. *$\widehat{\Phi}$ is c.e.*

Proof. Clearly Φ is c.e., since the sets $\Psi_{n,k}$ are uniformly computable. Let ϕ_m be a computable enumeration of the existential sentences of $\mathcal{L}$. We need only construct and algorithm for deciding, for each m, whether $\Phi \cup \{\, \phi_m \,\}$ is consistent.

For each $e \in \omega$, let $\mathcal{L}_e \subset \mathcal{L}$ be the language with symbols R_n and $Q_{n,k}$ for $\langle n, k\rangle \leq e$, and let $\Phi_e = \bigcup_{\langle n,k\rangle \leq e} \Psi_{n,k}$. Given ϕ_m, let e be least such that $\phi_m \in \mathcal{L}_e$. Suppose that $\Phi_e \cup \{\, \phi_m \,\}$ is consistent in $\mathcal{L}_e$ and let $\mathfrak{A}$ be a model for this set in $\mathcal{L}_e$. Let $\widehat{\mathfrak{A}}$ be the expansion of $\mathfrak{A}$ to $\mathcal{L}$ where $\widehat{\mathfrak{A}} \vDash \forall x, y\,(\neg x\ Q_{n,k}\ y)$ for all $\langle n, k\rangle > e$ and similarly for R_n. Since $\mathfrak{A}$ is the $\mathcal{L}_e$-reduct of $\widehat{\mathfrak{A}}$, they agree on all sentences of $\mathcal{L}_e$. Hence $\widehat{\mathfrak{A}} \vDash \Phi_e \cup \{\, \phi_m \,\}$.

We extend $\widehat{\mathfrak{A}}$ to a model $\mathfrak{B} \vDash \Phi$ as follows.

- The domain of $\mathfrak{B}$ is the domain of $\widehat{\mathfrak{A}}$ plus an extra element c.
- For all $\langle n, k\rangle$, and a, $b \in \mathfrak{B}$,

$$\mathfrak{B} \vDash (a\ Q_{n,k}\ b) \quad \Leftrightarrow \quad \begin{cases} \widehat{\mathfrak{A}} \vDash (a\ Q_{n,k}\ b) & \text{if } a,\ b \in \widehat{\mathfrak{A}} \\ a = b = c & \text{otherwise} \end{cases} \tag{4.18}$$

- For all $\langle n, k\rangle$, and a, $b \in \mathfrak{B}$, $\mathfrak{B} \vDash (a\ R_n\ b)$ if and only if $R_n \in \mathcal{L}_e$ and $\widehat{\mathfrak{A}} \vDash (a\ R_n\ b)$.

We will show that $\mathfrak{B} \vDash \Phi \cup \{\, \phi_m \,\}$.

As existential sentences are preserved upwards, we need only show that (4.17) holds for all $n \in Y_k$. Take any $n \in Y_k$ and $b \in \mathfrak{B}$ such that $\mathfrak{B} \vDash \exists y\,(b\ Q_{n,k}\ y)$. If $b = c$ then $\mathfrak{B} \vDash \forall z\,(\neg b\ R_n\ y)$ by construction. Otherwise, $b \in \widehat{\mathfrak{A}}$. So by (4.18), $Q_{n,k} \in \mathcal{L}_e$ and $\widehat{\mathfrak{A}} \vDash \exists y\,(b\ Q_{n,k}\ y)$. Then, as $\widehat{\mathfrak{A}} \vDash \Psi_{n,k}$, $\mathfrak{B} \vDash (\neg b\ R_n\ a)$ for all $a \in \widehat{\mathfrak{A}}$. So $\mathfrak{B} \vDash \forall z\,(\neg b\ R_n\ y)$ and hence (4.17) holds.

So, we have shown that $\Phi \cup \{\, \phi_m \,\}$ is consistent in $\mathcal{L}$ if and only if $\Phi_e \cup \{\, \phi_m \,\}$ is consistent in $\mathcal{L}_e$. $\Phi_e \cup \{\, \phi_m \,\}$ is a finite set of $\exists$ and $\forall$ sentences. By Theorem 4.2.3, we can decide its consistency uniformly in e and m. Hence $\widehat{\Phi}$ is c.e. ■

We can now define our theory T using $\widehat{\Phi}$ as a base. T will be the theory given by the following axioms.

(a) All elements of $\widehat{\Phi}$.

(b) $\forall x\, \exists y\,(x\ R_n\ y)$ for all $n \in X$.

As the elements of $\widehat{\Phi}$ are all either existential or universal, T is clearly $\forall_2$. To see that it is consistent, let $\mathfrak{A} \vDash \widehat{\Phi}$. We extend it to a model $\mathfrak{B} \vDash T$ as follows.

- The domain of $\mathfrak{B}$ is the domain of $\mathfrak{A}$ plus an extra element c.
- For all $\langle n, k\rangle$, and a, $b \in \mathfrak{B}$, $\mathfrak{B} \vDash (a\ Q_{n,k}\ b)$ if and only if a, $b \in \mathfrak{A}$ and $\mathfrak{A} \vDash (a\ Q_{n,k}\ b)$.
- For all n, and a, $b \in \mathfrak{B}$,

$$\mathfrak{B} \vDash (a\ R_n\ b) \quad \Leftrightarrow \quad \begin{cases} \mathfrak{A} \vDash (a\ R_n\ b) & \text{if } a,\ b \in \mathfrak{A} \\ b = c \text{ and } n \in X & \text{otherwise} \end{cases} \tag{4.19}$$

By construction, $\mathfrak{B} \models \forall x\,(x\ R_n\ c)$. So, as usual, we need only show that (4.17) holds for all $n \in Y_k$. Take any $n \in Y_k$ and $b \in \mathfrak{B}$ such that $\mathfrak{B} \models \exists y\,(b\ Q_{n,k}\ y)$. Then, by construction, $b \in \mathfrak{A}$ and $\mathfrak{A} \models \forall z\,(\neg b\ R_n\ z)$. Hence $\mathfrak{B} \models (\neg b\ R_n\ a)$ for all $a \neq c$. Furthermore, since $n \in Y_k$, $n \notin X$ and hence $\mathfrak{B} \models (\neg b\ R_n\ c)$. So $\mathfrak{B} \models \forall z\,(\neg b\ R_n\ y)$ and thus (4.17) holds.

So T is a consistent $\forall_2$ theory. It is also $\forall$ decidable since $\widehat{\Phi}$ is c.e. and 1-complete. Suppose now for a contradiction that $\widehat{T} \supseteq T$ is a consistent c.e. extension. For all $n \in Y_k$, $\Psi_{n,k} \subseteq \widehat{T}$ implies that

$$\widehat{T} \vdash \exists x\,\forall y\,(\neg x\ R_n\ y) \tag{4.20}$$

So we have that

$$\widehat{T} \vdash \begin{cases} \forall x\,\exists y\,(x\ R_n\ y) & \text{if } n \in X \\ \exists x\,\forall y\,(\neg x\ R_n\ y) & \text{if } n \notin X \end{cases} \tag{4.21}$$

As $\widehat{T}$ is c.e., these means that X is computable, which is a contradiction. □

4.3 Characterizations for computable models

If we want to extend Theorem 4.2.5 to a characterization for the existence of a computable e.c. model, we need to find a necessary condition for their existence. We can do this will the help of the following result.

Proposition 4.3.1. *Let T be a consistent, $\forall$-c.e. $\forall_2$ theory. Let $\mathfrak{A} \models T$ be computable and existentially closed in T. Then $\mathfrak{A}$ is 1-decidable.*

Proof. We need to show that the predicate $\mathfrak{A} \models \psi(\overline{a})$ is decidable for $\exists$ formulae ψ and $\overline{a} \in \mathfrak{A}$. The set of all existential $\psi(\overline{a})$ such that $\mathfrak{A} \models \psi(\overline{a})$ is clearly c.e.; we just enumerate the elements of $\mathfrak{A}$ until we find witnesses for $\psi(\overline{a})$. So we need only show that the following set is c.e.

$$\mathrm{Th}_\forall(\mathfrak{A}, \mathrm{dom}(\mathfrak{A})) = \{\, \varphi(\overline{a}) \mid \mathfrak{A} \models \varphi(\overline{a}),\ \varphi(\overline{x}) \text{ is universal, and } \overline{a} \in \mathfrak{A} \,\} \tag{4.22}$$

Since $\mathfrak{A}$ is e.c., we know from Theorem 2.2.3 that $\varphi(\overline{a}) \in \mathrm{Th}_\forall(\mathfrak{A}, \mathrm{dom}(\mathfrak{A}))$ if and only if there is an $\exists$ formula ψ satisfying (4.13) such that $\mathfrak{A} \models \psi(\overline{a})$. So, to enumerate $\mathrm{Th}_\forall(\mathfrak{A}, \mathrm{dom}(\mathfrak{A}))$, we just enumerate the $\exists$ formula ψ such that $\mathfrak{A} \models \psi(\overline{a})$, and use the fact that T is $\forall$-c.e. to search for some universal φ satisfying (4.13). □

This proposition gives us our first characterization for the existence of computable models.

Theorem 4.3.2. *Let T be an $\forall$-c.e. $\forall_2$ theory. Then T has a computable, existentially closed model if and only if it has a consistent, $\forall$-decidable 1-completion $\widehat{T}$ such that for all universal sentences φ*

$$\widehat{T} \vdash \varphi \Rightarrow \text{ there is an existential } \psi \text{ with } \widehat{T} \vdash \psi,\ T \vdash (\psi \to \varphi) \tag{4.23}$$

Proof. ($\Rightarrow$) Let $\mathfrak{A}$ be a computable, e.c. model of T. Let $\widehat{T} = \mathrm{Th}_{\forall_2}(\mathfrak{A})$. $\widehat{T}$ is clearly a 1-complete extension of T. It has the property in (4.23) as this is just a special case of property (b) from Theorem 4.2.5. Finally, it is $\forall$-decidable since $\mathfrak{A}$ is 1-decidable by Proposition 4.3.1.

($\Leftarrow$) By Theorem 4.2.5, there is some computable, e.c. model $\mathfrak{A} \models \widehat{T}$. However, this does not necessarily meant that $\mathfrak{A}$ is existentially closed in T. This is exactly what happens in Example 4.1.4. The model $\mathfrak{A}_1$ with a single element is decidable and e.c. in $\mathrm{Th}_{\forall_2}(\mathfrak{A}_1)$, but it is not e.c. in the original theory.

To show that $\mathfrak{A}$ is existentially closed in T, we again appeal to the characterization of e.c. models in Theorem 2.2.3. Take any $\overline{a} \in \mathfrak{A}$ and $\forall$ formula φ such that $\mathfrak{A} \models \varphi(\overline{a})$. Then there is some $\exists$ formula ψ such that $\mathfrak{A} \models \psi(\overline{a})$ and

$$\widehat{T} \vdash \forall \overline{x} \left(\psi(\overline{x}) \rightarrow \varphi(\overline{x})\right) \tag{4.24}$$

The sentence in (4.24) is universal. Therefore, there is some existential sentence χ such that $\mathfrak{A} \models \chi$ and

$$T \vdash \left(\chi \rightarrow \forall \overline{x} \left(\psi(\overline{x}) \rightarrow \varphi(\overline{x})\right)\right) \tag{4.25}$$

So then, $\mathfrak{A} \models (\chi \wedge \psi(\overline{a}))$ and

$$T \vdash \forall \overline{x} \left((\chi \wedge \psi(\overline{x})) \rightarrow \varphi(\overline{x})\right) \tag{4.26}$$

This implies that $\mathfrak{A}$ is e.c. in T. $\square$

Like Theorem 4.1.2, this characterization has a rather unfortunate feature; we were forced to add an extra effectiveness condition, namely that the theories must be $\forall$-c.e. Once again, this does not capture all of the $\forall_2$ theories with computable e.c. models. Consider the theory $\mathrm{Th}_{\forall_2}(\mathbb{N})$ from Example 4.1.3. As $\mathrm{Th}_{\forall_2}(\mathbb{N})$ is 1-complete and all of the elements of $\mathbb{N}$ are represented by closed terms in this language, $\mathbb{N}$ is e.c. in $\mathrm{Th}_{\forall_2}(\mathbb{N})$. However, the list of all $\forall$ setences true of $\mathbb{N}$ is Π^0_1, not c.e.

Furthermore, Theorem 4.1.2 characterizes some theories that Theorem 4.3.2 does not. The theory $\mathrm{Th}_1(\mathbb{N})$ has a Δ^0_2 list of axioms, namely all of the existentials and universals true of $\mathbb{N}$. However, it is also not c.e. While it may be possible that Theorem 4.3.2 characterizes some theories that Theorem 4.1.2 does not, this is still unknown. In particular, it is still unknown whether there are any $\forall$-c.e. $\forall_2$ theories which can not be extended to an $\forall_2$ theory with a Δ^0_2 list of axioms.

However, Theorem 4.3.2 is significant because it is a characterization for the existence of computable e.c. models, and not just computable models. Furthermore, since the computable e.c. models characterized are always 1-decidable, we have the following related characterization.

Corollary 4.3.3. *Let T be an $\forall_2$-theory. Then T has a* 1*-decidable e.c. model if and only if it has a consistent,* $\forall$*-decidable* 1*-completion $\widehat{T}$ satisfying* (4.23).

Proof. ($\Rightarrow$) Let $\mathfrak{A}$ be a 1-decidable e.c. model of T. As with Theorem 4.3.2, let $\widehat{T} = \mathrm{Th}_{\forall_2}(\mathfrak{A})$. This satisfies (4.23) as before.

($\Leftarrow$) By Theorem 4.2.5 and Proposition 4.3.1 there is a 1-decidable e.c. model $\mathfrak{A} \models \widehat{T}$. As with Theorem 4.3.2, (4.23) ensures that $\mathfrak{A}$ is existentially closed in T. □

This result is a much better analogue of the effective completeness theorem than Theorem 4.1.2. It is a characterization for all $\forall_2$ theories, and not just those with certain effective conditions. Furthermore, it is a characterization for the existence 1-decidable e.c. models, further supporting our claim that there is a correspondence between e.c. models of $\forall_2$ theories and models of complete theories.

As a final word, we should note that Theorem 4.3.2 also gives us the following characterization.

Corollary 4.3.4. *Let T be an $\forall_2$-theory. Then T has a 1-decidable model if and only if it has a consistent, $\forall$-decidable 1-completion $\widehat{T}$.*

Proof. This is simply an observation that we only needed (4.23) to ensure that the property of existential closure was preserved between the two theories. □

This result is interesting on its own. As Example 4.2.6 demonstrates, there are $\forall_2$ theories with a 1-decidable model but no decidable model. Hence, this result is a nice analogue of Theorem 3.1.1 for the existence of 1-decidable models of $\forall_2$ theories.

4.4 Generalizing to $\forall_n$ theories

As we saw in Proposition 2.5.3, we can often relativize results about $\forall_2$ theories to more complicated theories. In particular, we can use the results of the previous section to construct n-decidable models for $\forall_{n+1}$ theories.

Theorem 4.4.1. *Let T be a consistent, $\forall_n$-decidable $\forall_{n+1}$ theory, $n \geq 1$. Then T has an n-decidable model.*

Proof. We proceed by induction on n. For the case $n = 1$, we know that T has a computable e.c. model by Theorem 4.2.5. This model is 1-decidable by Proposition 4.3.1.

Suppose that our claim holds for some $n = k \geq 1$. We will use the technique from Proposition 2.5.3 to show that it holds for $n = k + 1$. We introduce a new relation R_φ for each $\exists$ formula $\varphi(\overline{x}) \in \mathcal{L}$. Assuming that k is even, we let $\widehat{T}$ be the theory in $\mathcal{L}(R_\varphi)_\varphi$ given by the following list of axioms.

(a) $\forall\overline{x}\big(\varphi(\overline{x}) \leftrightarrow R_\varphi(\overline{x})\big)$ if φ is an $\exists$ formula.

(b) $\forall\overline{x} \ldots \forall\overline{y}\, R_{\varphi(\overline{x},\ldots,\overline{y})}(\overline{x},\ldots,\overline{y})$ for all $\forall_{k+2}$ sentences $\forall\overline{x} \ldots \forall\overline{y}\, \varphi(\overline{x},\ldots,\overline{y}) \in T$, with φ is existential.

In the case that k is odd, we replace the axioms in (b), with the axioms

(b)' $\forall \overline{x} \ldots \exists \overline{y} \, \neg R_{\neg\varphi(\overline{x},\ldots,\overline{y})}(\overline{x}, \ldots, \overline{y})$ for all $\forall_{k+2}$ sentences $\forall \overline{x} \ldots \exists \overline{y} \, \varphi(\overline{x}, \ldots, \overline{y}) \in T$, with φ is existential.

In either case, $\widehat{T}$ is a extension of T, since $\widehat{T} \vdash \psi$ for all $\forall_{n+1}$ sentences $\psi \in T$. The axioms (b) and (b)' ensure that $\widehat{T}$ is $\forall_{k+1}$. Finally, $\widehat{T}$ is consistent by exactly the same argument as Proposition 2.5.3.

For any formula $\varphi \in \mathcal{L}(R_\varphi)_\varphi$, we define $\overline{\varphi}$ by induction.

- $\overline{\xi} = \xi$ for any atomic formula $\xi \in \mathcal{L}$.
- $\overline{(R_\varphi(\overline{x}))} = \varphi(\overline{x})$ for all $\exists$ formulae φ.
- $\overline{(\psi \wedge \chi)} = \overline{\psi} \wedge \overline{\chi}$.
- $\overline{(\neg\psi)} = \neg\overline{\psi}$.
- $\overline{(\exists \overline{x} \, \psi(\overline{x}))} = \exists \overline{x} \, \overline{\psi(\overline{x})}$.

Lemma 4.4.2. *For any $\forall_k$ sentence $\varphi \in \mathcal{L}(R_\varphi)_\varphi$, $\overline{\varphi}$ is an $\forall_{k+1}$ sentence of $\mathcal{L}$. Furthermore, $\widehat{T} \vdash \varphi$ if and only if $T \vdash \overline{\varphi}$.*

Proof. This follows from a straightforward induction on the complexity of φ and the definition of $\overline{\varphi}$. ☐

As T is $\forall_{k+1}$-decidable, this lemma shows that $\widehat{T}$ is $\forall_k$-decidable. So, by our inductive hypothesis, $\widehat{T}$ has a k-decidable model $\widehat{\mathfrak{A}}$. Let $\mathfrak{A}$ be the reduction of $\widehat{\mathfrak{A}}$ to $\mathcal{L}$. We need only show that $\mathfrak{A}$ is $(k+1)$-decidable.

We assume that k is even; the proof for when k is odd is analogous. Take any $\forall_{k+1}$ formula $\varphi(\overline{x})$ where $\varphi(\overline{x}) = \forall \overline{y} \ldots \forall \overline{z} \, \psi(\overline{x}, \overline{y}, \ldots, \overline{z})$ for some existential formula ψ. Then, for any $\overline{a} \in \mathfrak{A}$,

$$\mathfrak{A} \vDash \forall \overline{y} \ldots \forall \overline{z} \, \psi(\overline{a}, \overline{y}, \ldots, \overline{z}) \quad \Leftrightarrow \quad \widehat{\mathfrak{A}} \vDash \forall \overline{y} \ldots \forall \overline{z} \, R_{\varphi(\overline{x},\overline{y},\ldots,\overline{z})}(\overline{a}, \overline{y}, \ldots, \overline{z}) \tag{4.27}$$

So $\mathfrak{A}$ is $(k+1)$-decidable since $\widehat{\mathfrak{A}}$ is k-decidable. ☐

Since any n-decidable model corresponds to an $\forall_n$-decidable n-completion, this result allows us to extend Corollary 4.3.4 to a characterization of the existence of n-decidable models.

Corollary 4.4.3. *Let T be an $\forall_{n+1}$-theory. Then T has a n-decidable model if and only if it has a consistent, $\forall_n$-decidable n-completion $\overline{T}$.*

Proof. This proof is identical that of Corollary 4.3.4. ☐

A similar relativization of Theorem 4.1.2 yields the following result.

Theorem 4.4.4. *Let T be a theory with a Δ^0_2 list of $\forall_{n+1}$ axioms, for $n \geq 1$. Then T has a $(n-1)$-decidable model if and only if T has a consistent n-completion $\widehat{T}$ such that $\widehat{T}$ is $\exists_n$-c.e.*

Proof. Again, we proceed by induction on n. The case $n = 1$ is just Theorem 4.1.2. Suppose it is true for $n = k$. The set of $\exists_{k+1}$ sentences true in an k-decidable model is clearly c.e. So, to show that this claim holds for $n = k+1$, we need only show that, if T is an $(k+1)$-complete, $\exists_{k+1}$-c.e. theory with a Δ^0_2 list of $\forall_{k+2}$ axioms, then it has an k-decidable model.

Will we assume that k is even; as usual, the case when k is odd will be analogous. We let $\widehat{T}$ be the relativization of T from Theorem 4.4.1, and $\overline{\varphi}$ the transformation from $\mathcal{L}(R_\varphi)_\varphi$ to $\mathcal{L}$. As with Lemma 4.4.2, we can show that $\overline{\varphi}$ is an $\exists_{k+1}$ sentence of $\mathcal{L}$ whenever φ is an $\exists_k$ sentence of $\mathcal{L}(R_\varphi)_\varphi$. So $\widehat{T}$ is $\exists_k$-c.e. since T is $\exists_{k+1}$-ce. Similarly, it is k-complete since T is $(k+1)$-complete. Finally, if Φ is a Δ^0_2 list of $\forall_{k+2}$ axioms for T then

$$\begin{aligned}\widehat{\Phi} = &\left\{\, \forall\overline{x}\big(\varphi(\overline{x}) \leftrightarrow R_\varphi(\overline{x})\big) \mid \varphi \text{ is an } \exists_k \text{ formula} \,\right\} \cup \\ &\left\{\, \forall\overline{x} \ldots \forall\overline{y}\, R_{\varphi(\overline{x},\ldots,\overline{y})}(\overline{x},\ldots,\overline{y}) \mid \forall\overline{x} \ldots \forall\overline{y}\, \varphi(\overline{x},\ldots,\overline{y}) \in \Phi \,\right\}\end{aligned} \tag{4.28}$$

is a Δ^0_2 list of $\forall_{k+1}$ axioms for $\widehat{T}$.

So this theory has a $(k-1)$-decidable model $\widehat{\mathfrak{A}}$ by our inductive hypothesis. As with Theorem 4.4.1, the reduction $\mathfrak{A}$ of $\widehat{\mathfrak{A}}$ to $\mathcal{L}$ is k-decidable since $\widehat{\mathfrak{A}}$ is $(k-1)$-decidable. $\square$

An unfortunate feature of both of these results is that, as we increased the quantifier complexity of the theory, we had to increase its effectiveness, hence increasing the decidability of the constructed model. We would prefer to restrict our characterizations for $\forall_n$ theories to computable, or even 1-decidable, models, just as we did with $\forall_2$ theories. However, as the next example shows, we must increase the effectiveness of the theory as we increase its quantifier complexity.

Example 4.4.5. There is a consistent c.e., 1-complete, $\exists_2$ theory with no computable model.

Proof. Our language $\mathcal{L}$ will consist of a binary relation R and infinitely many unary relations P_n. Let X and Y be a pair of computably inseparable c.e. sets. For finite $F \subset A$, $G \subset B$, define $\delta_{F,G}(x,y)$ by

$$\delta_{F,G}(x,y) =_{\text{def}} \bigwedge_{n\in F} ((x\ R\ y) \wedge P_n(x)) \wedge \bigwedge_{m\in G} ((x\ R\ y) \wedge \neg P_m(x)) \tag{4.29}$$

Our theory T is given by the following list of axioms.

- All consistent existential sentences in our language $\mathcal{L}$.

- We want a unique element R-connected to all other elements. We can guarantee this by the sentence

$$\exists x, z \left(\forall y \,(x\ R\ y) \wedge \forall w \,(w\ R\ z \rightarrow x = w)\right) \tag{4.30}$$

- The unique element R-connected to all others defines a separation of A and B. We express this by the sentences $\exists x\, \forall y\, \delta_{F,G}(x, y)$ for all finite $F \subset A$, $G \subset B$.

Lemma 4.4.6. *T is a consistent, c.e. $\exists_2$ theory.*

Proof. This theory is clearly $\exists_2$, and it is c.e. since both X and Y are. To see that it is consistent, let Ψ be the set of all consistent existential sentences in our language $\mathcal{L}$. Following the proof of Lemma 4.2.7 in Example 4.2.6, we see that Ψ is consistent. Let $\mathfrak{A} \models \Psi$.

We extend $\mathfrak{A} \models \Psi$ to a model $\mathfrak{B} \models T$ as follows.

- The domain of $\mathfrak{B}$ is the domain of $\mathfrak{A}$ plus an extra element c.
- For all a, $b \in \mathfrak{B}$,

$$\mathfrak{B} \models (a\ R\ b) \quad \Leftrightarrow \quad \begin{cases} \mathfrak{A} \models (a\ R\ b) & \text{if } a,\, b \in \mathfrak{A} \\ a = c & \text{otherwise} \end{cases} \tag{4.31}$$

- For all $a \in \mathfrak{B}$,

$$\mathfrak{B} \models P_n(a) \quad \Leftrightarrow \quad \begin{cases} \mathfrak{A} \models P_n(a) & \text{if } a \in \mathfrak{A} \\ n \in A & \text{otherwise} \end{cases} \tag{4.32}$$

By construction, $\mathfrak{B} \models \forall y\,(c\ R\ y)$. Furthermore, for any $a \in \mathfrak{B}$, $\mathfrak{B} \models (a\ R\ c)$ if and only if $a = c$. So $\mathfrak{B}$ satisfies (4.30). Finally, as A and B are disjoint, $\mathfrak{B} \models \forall y\, \delta_{F,G}(c, y)$ for all finite $F \subseteq A$, $G \subseteq B$. So $\mathfrak{B} \models T$. □

Lemma 4.4.7. *Let $\mathfrak{A} \models T$. Then there is a unique $\overline{a} \in \mathfrak{A}$ such that $\mathfrak{A} \models \forall y\,(a\ R\ y)$.*

Proof. Let $a \in \mathfrak{A}$ be such that

$$\mathfrak{A} \models \exists z \left(\forall y\,(a\ R\ y) \wedge \forall w\,(w\ R\ z \rightarrow a = w)\right) \tag{4.33}$$

Clearly, $\mathfrak{A} \models \forall y\,(a\ R\ y)$. Furthermore, there is some element $b \in \mathfrak{A}$ such that for all $c \in \mathfrak{A}$, $\mathfrak{A} \models ((c\ R\ b) \rightarrow c = a)$. If $\mathfrak{A} \models \forall y\,(c\ R\ y)$, then $\mathfrak{A} \models (c\ R\ b)$. So $a \in \mathfrak{A}$ is unique. □

Suppose now for a contradiction that $\mathfrak{A}$ is a computable model of T. Let $a \in \mathfrak{A}$ be the unique element such that $\mathfrak{A} \models \forall y\,(a\ R\ y)$. Take any finite $F \subseteq A$, $G \subseteq B$. $\mathfrak{A} \models \exists x\,\forall y,\ \delta_{F,G}(x,y)$, and hence $\mathfrak{A} \models \forall y,\ \delta_{F,G}(a,y)$ by the uniqueness of $a \in \mathfrak{A}$. As this is true fo all finite $F \subseteq A$, $G \subseteq B$,

$$\mathfrak{A} \models \begin{cases} P_n(a) & \text{if } n \in A \\ \neg P_n(a) & \text{if } n \in B \end{cases} \tag{4.34}$$

As $\mathfrak{A}$ is computable, a defines a computable separation of A and B, which is a contradiction. □

Corollary 4.4.8. *There is a consistent Δ^0_2, $\forall$-decidable, 2-complete $\exists_2$ theory with no computable model.*

Proof. The question of whether or not a c.e. set of sentences Φ is consistent is uniformly computable in $\mathbf{0}'$. So we can create a Δ^0_2 sequence of c.e. extensions of the theory T from Example 4.4.5 by adding a formula at a time and testing for consistency. The union $\widehat{T}$ of these extensions is consistent, Δ^0_2, and 2-complete. It is also $\forall$-decidable since it agrees with T on all universal sentences. □

We see from Corollary 4.4.8 that if we want to characterize the existence of computable models of $\exists_2$ theories, we need effectiveness conditions that are different from the ones we have already defined. If an $\exists_2$ theory has a consistent $\exists_2$-c.e. 2-completion then it has a 1-decidable model by Theorem 4.4.4. However, if we allow it to have only an extension where the set of $\exists_2$ consequences is Δ^0_2, and all the $\exists$ and $\forall$ consequences are decidable, then we can no longer guarantee even a computable model. Therefore, any characterization for the existence of computable models of these theories will require something other than the position of the set of $\exists_n (\forall_n)$ consequences in the Kleene heirarchy.

4.5 Computable prime models

In Chapter 2 we saw that existentially closed models of $\forall_2$ theories could have additional properties, such as being algebraically prime. An interesting question is whether we can characterize when there is a computable model with this property. This question is motivated by a similar result of Harrington [20] for decidable elementarily prime models.

Theorem 4.5.1 (Harrington [20]). *For any theory T, the following properties are equivalent.*

(a) T has a decidable elementarily prime model.

(b) T has an elementarily prime model and the principal, complete types of T are uniformly computable.

As we illustrated in Chapter 2, there is a strong correspondence between elementarily prime models of complete theories and e.c., algebraically prime models of 2-complete theories. Furthermore, as we saw in Corollary 4.3.4, there is a similarity between the conditions for the existence of a decidable model of a theory, and the conditions for the existence of a 1-decidable model of an $\forall_2$ theory. Together, these relationships suggest the following result.

Theorem 4.5.2. *Let T be a 2-complete $\forall_2$ theory with an e.c., algebraically prime model. If the isolated, maximal $\exists$ types of T are uniformly computable, then T has a 1-decidable, e.c., algebraically prime model.*

Proof. This proof is essentially in the style of the argument for Theorem 4.5.1 given by Harrington. Let Γ_e be a uniformly computable enumeration of the isolated, maximal $\exists$ types of T. As these $\exists$ types are each realized in some model of T, let $\mathfrak{A}$ be a model realizing Γ_0. As T is 2-complete and Γ_0 is maximal, for any universal sentence φ,

$$T \vdash \varphi \Leftrightarrow \mathfrak{A} \models \varphi \Leftrightarrow \varphi \notin \Gamma \tag{4.35}$$

So T is $\forall$-decidable since Γ_0 is computable.

As with effective model theoretic forcing, we will use $\forall$-decidability to construct a computable set of atomic formula p_ω. However, we will not be able to rely on Theorem 3.3.5 for our construction. This is because an arbitrary effective strategy for making $\overline{\mathfrak{A}}$ computable may introduce some maximal $\exists$ types that are not isolated. So we will have to give a more careful construction of p_ω.

Even though this argument will not involve effective model theoretic forcing *per se*, we will use much the same terminology in our construction. We will construct a computable sequence of elements $p_s \in N_T$ with $p_\omega = \bigcup_s p_s$. Provided that p_ω is complete for atomic sentences in $\mathcal{L}(W)$, Proposition 3.3.1 will guarantee that the canonical model for p_ω is computably presentable. As usual, we will denote this model as $\overline{\mathfrak{A}}$.

Our construction will be a finite injury argument. Let θ_n be an enumeration of the atomic sentences, and τ_n be an enumeration of the closed terms of $\mathcal{L}(W)$. Furthermore, for convenience, we will assume that $W = \omega$, and for each $n \in \omega$, $\vec{n}$ is the initial segment of ω of length n. Then we will construct p_ω so that it satisfies the following requirements.

(P_k) $\theta_k \in p_\omega$ or $\neg\theta_k \in p_\omega$.

(Q_k) $(\tau_k = n) \in p_\omega$ for some $n \in \omega$.

(R_n) There is an $e \in \omega$ such that $\exists\text{-type}_{\overline{\mathfrak{A}}}(\vec{n}) = \Gamma_e$.

As the following result shows, these requirements are enough.

Lemma 4.5.3. *If p_ω is computable and satisfies all of the above requirements, then the compiled structure $\overline{\mathfrak{A}}$ is computable, and an algebraically prime model of T.*

Proof. The requirements (P_e) guarantee that $\overline{\mathfrak{A}}$ is computable. Furthermore, since $p_s \in N_T$ for each s, we know from the proof of Theorem 4.2.5 that $\overline{\mathfrak{A}} \models T_\forall$.

From Proposition 3.2.3 we see that requirement (Q_e) ensures that every element $a \in \overline{\mathfrak{A}}$ is of the form $n^{\overline{\mathfrak{A}}}$ for some $n \in \omega$. Take any sequence of elements $\overline{a} \in \overline{\mathfrak{A}}$ and let $\vec{n}$ be an initial segment such that each b in $\overline{a}$ is of the form $k^{\overline{\mathfrak{A}}}$ for some k in $\vec{n}$. As (R_n) holds, there is some e such that $\exists\text{-type}_{\overline{\mathfrak{A}}}(\vec{n}) = \Gamma_e$. As T is 2-complete, we know from Proposition 2.4.13 that Γ_e has some $\exists$ support φ. Then as $\exists\text{-type}(\overline{a}) \subseteq \Gamma_e$, we can get an $\exists$ support for $\exists\text{-type}(\overline{a})$ by existentially quantifying out the variables of φ that do not correspond to the elements of $\overline{a}$.

Hence we have that $\exists\text{-type}(\overline{a})$ is maximal for every $\overline{a} \in \overline{\mathfrak{A}}$. So $\overline{\mathfrak{A}} \models T$ by Theorem 2.3.4. Furthermore, we have shown that every $\exists$ type realized in $\overline{\mathfrak{A}}$ has $\exists$ support, so $\overline{\mathfrak{A}}$ is $\exists$-atomic. Therefore $\overline{\mathfrak{A}}$ is e.c. and algebraically prime by Theorem 2.4.10. ∎

So we need only construct $p_\omega = \bigcup_s p_s$ satisfying the above requirements. At stage s, for $n \in \omega$, we define

$$\delta^s_{\vec{n}}(\overline{x}) = \bigwedge_{\rho(\vec{n},\overline{m}) \in p_s} \exists \overline{y} \rho(\overline{x}, \overline{y}) \tag{4.36}$$

Also at stage s, we let $e^s_{\vec{n}}$ be the candidate for e satisfying requirement (R_n). We say that requirement (R_n) *is injured* at stage s if

$$\delta^s_{\vec{n}}(\overline{x}) \notin \Gamma_{e^s_{\vec{n}}}(\overline{x}) \tag{4.37}$$

We will be able to determine whether or not a requirement is injured because the Γ_e are uniformly computable and not just c.e.

Stage $s = -1$:
At the initial stage, we set $p_{-1} = \emptyset$, and $e^{-1}_{\vec{n}} = -1$ for all $\vec{n} \in W$.

Stage $s = 3k$:
At this stage, we ensure that requirement (P_k) is met. As N_T is complete for atomic sentences, we add either θ_k or $\neg\theta_k$ to p_{s-1} so as to injure the requirement of least possible priority. For all $k \le s$, we know from Theorem 2.4.14 that there is some e such that $\delta^s_{\vec{k}} \in \Gamma_e$. For each $n \le s$ such that (R_n) was injured, or $e^{s-1}_{\vec{n}} = -1$, we let $e^s_{\vec{n}}$ be the least such element.

Stage $s = 3k + 1$:
At this stage, we ensure that requirement (Q_k) is met. Let $m \in \omega$ be the least element of W not appearing in p_{s-1} or τ_k. As N_T is a notion of forcing, we can add $(\tau_k = m)$ to p_{s-1}. We leave $e^s_{\vec{n}}$ untouched for all $n \in \omega$.

Stage $s = 3\langle n, i\rangle + 2$:
At this stage, we ensure that requirement (R_n) is met. Let $\varphi(\overline{x})$ be the least $\exists$ formula less than s in $\Gamma_{e^s_{\vec{n}}}$ not yet witnessed in p_{s-1}. Using the techniques of

Proposition 4.2.4, we extend p_{s-1} to witness $\varphi(\vec{n})$ in a way such that no requirements of equal or higher priority are injured. If it is not possible to extend p_{s-1} this way, then set $p_s = p_{s-1}$ and let $e^s_{\vec{n}}$ be the least element such that $\delta^s_{\vec{n}} \in \Gamma_{e^s_{\vec{n}}}$.

Either way, for any $k \leq s$ such that (R_k) is injured, or $e^{s-1}_{\vec{k}} = -1$, we let $e^s_{\vec{m}}$ be the least element such that $\delta^s_{\vec{m}} \in \Gamma_{e^s_{\vec{m}}}$.

Clearly the requirements (P_k) and (Q_k) are all met. We need only show that each requirement (R_n) is injured at most finitely often and is eventually met. In fact, we will show a stronger claim; for every $n \in \omega$, there is a stage s_n such that

(a) (R_n) is never injured after stage s_n and hence $\Gamma_{e^{s_n}_{\vec{n}}} = \exists\text{-type}(\vec{n})$.

(b) $\delta^{s_n}_{\vec{n}}$ is an $\exists$ support of the maximal $\exists$ type $\Gamma_{e^{s_n}_{\vec{n}}}$.

Take any $n \in \omega$ and suppose these properties hold for all requirements (R_k) of higher priority. Let $t > n$ be the least stage greater than all the s_k. By construction, $\delta^t_{\vec{n}}(\overline{x}) \in \Gamma_{e^t_{\vec{n}}}$. For each $k < n$, $\delta^{s_k}_{\vec{k}}(\overline{y})$ is just $\delta^t_{\vec{n}}(\overline{x})$ with some of the variables bound by an existential quantifier. So we have that $\delta^{s_k}_{\vec{k}}(\overline{y}) \in \Gamma_{e^t_{\vec{n}}}$ for all $k < n$. This means that $\Gamma_{e^{s_k}_{\vec{k}}} \subseteq \Gamma_{e^t_{\vec{n}}}$ for all $k < n$. So, for each stage $s \geq t$, we can always choose some extension of p_s which does not injure (R_n).

As (R_n) is never injured after stage t, the stages $s = 3\langle n, i\rangle + 2$ of our construction guarantee that $\exists\text{-type}(\vec{n}) = \Gamma_{e^t_{\vec{n}}}$, an isolated, maximal $\exists$ type. Let φ be an $\exists$ support of $\exists\text{-type}(\vec{n})$. Then eventually there is some stage $s \geq t$ such that $p \vdash \varphi(\vec{n})$. So $\delta^s_{\vec{n}}$ is an $\exists$ support of $\exists\text{-type}(\vec{n})$, and hence we set $s_n = s$. ☐

This result gives us the following analogue of Theorem 4.5.1.

Corollary 4.5.4. *Let T be a 2-complete $\forall_2$ theory. The following are equivalent.*

(a) T has a 1-decidable, e.c., algebraically prime model.

(b) T has an e.c. algebraically prime model and the isolated, maximal $\exists$ types of T are uniformly computable.

Proof. (a) $\Rightarrow$ (b) Let $\mathfrak{A}$ be a 1-decidable, e.c., algebraically prime model. Let Γ be any isolated, maximal $\exists$ type. As T is 2 complete, Γ has some $\exists$ support $\varphi(\overline{x})$ and there is some $\overline{a} \in \mathfrak{A}$ such that $\mathfrak{A} \models \varphi(\overline{a})$. So $\mathfrak{A}$ realizes Γ. Hence the isolated, maximal $\exists$ types of T are exactly the types of the form $\exists\text{-type}(\overline{a})$ for some $\overline{a} \in \mathfrak{A}$. As $\mathfrak{A}$ is 1-decidable, we can compute these uniformly.

(b) $\Rightarrow$ (a) This direction is just Theorem 4.5.2. ☐

Given Theorem 2.4.16, we can easily extend Corollary 4.5.4 to the following.

Corollary 4.5.5. *Let T be a complete $\forall_2$ theory. The following are equivalent.*

(a) T has a 1-decidable elementarily prime model.

(b) T has an elementarily prime model and the isolated, maximal $\exists$ types of T are uniformly computable.

As usual, we would like that characterize the existence of computable models with these properties, not just 1-decidable ones. Again, as in Theorem 4.3.2, we can do this provided that we have an extra effectiveness condition on our theory.

Corollary 4.5.6. *Let T be a 2-complete, $\forall$-c.e. $\forall_2$ theory. The following are equivalent.*

(a) T has a computable, e.c., algebraically prime model.

(b) T has an e.c. algebraically prime model and the isolated, maximal $\exists$ types of T are uniformly computable.

Proof. We need only show that (a) $\Rightarrow$ (b). This follows from the fact that any computable, e.c. model of T is 1-decidable. ☐

A similar result holds for complete, $\forall$-c.e. $\forall_2$ theories and computable elementarily prime models.

4.6 Omitting types in computable models

Another model theoretic property that we considered in Chapter 2 was that of omitting a set of formulae. As we saw in Theorem 3.3.7, there is an effective omitting types theorem for decidable models. Our results from section 4.5 give us similar effective versions of the results from Chapter 2.

Theorem 4.6.1. *Let T be a consistent $\forall$-decidable $\forall_2$ theory. Let Φ_n be a Σ^0_2 list of c.e. $\forall$ sets, none of which have $\exists$ support. Then we can effectively enforce a model $\overline{\mathfrak{A}} \models T$ omitting all the Φ_n.*

Proof. Let Γ_e be a computable enumeration of all c.e. $\forall$ sets over $\mathcal{L}$ and fix some $e \in \omega$. Let $\overline{c}$ be any sequence of elements in W with $|\overline{c}| = |\overline{x}|$. Given $p \in N_T$, we search for some universal formula $\varphi(\overline{x}) \in \Gamma_e$ such that $T \cup p \cup \{ \neg\varphi(\overline{c}) \}$ is consistent and extend p accordingly.

This procedure is not computable in general, because there may not be any such $\varphi(\overline{x})$. However, as with the proof of Theorem 2.5.2 given in Keisler [30], there is such a $\varphi \in \Gamma_e$ if Γ_e does not have $\exists$ support. So the property that $\overline{c}$ does not realize Γ_e in $\overline{\mathfrak{A}}$ is uniformly enforceable over $\overline{c} \in W$ and the Σ^0_2 list Φ_n. So we are done by Theorem 3.3.9. ☐

Corollary 4.6.2. *Let T be a consistent, c.e., 2-complete $\forall_2$ theory. Let Φ_n be a Σ^0_2 list of c.e. $\forall_2$ sets, none of which have $\exists$ support. Then we can effectively enforce a model $\overline{\mathfrak{A}} \models T$ omitting all the Φ_n.*

Proof. This proof uses the relativization that we used in Proposition 2.5.3 and Theorem 4.4.1. As usual, we introduce a new relation R_φ for each $\exists$ formula $\varphi(\overline{x}) \in \mathcal{L}$, and let $\widehat{T}$ be defined as before. Since T is c.e. and 2-complete, it is $\forall_2$-decidable. So, as we say in the proof of Theorem 4.4.1, $\widehat{T}$ is $\forall$-decidable. □

These are not the only kind of omitting types theorems that we can have for models of $\forall_2$ theories. An interesting variation on these types of theorems is the existence of an $\exists$-minimal pair.

Definition 4.6.3. Two structures $\mathfrak{A}_0$, $\mathfrak{A}_1$ are an $\exists$-*minimal pair* if every maximal $\exists$ type realized in both $\mathfrak{A}_0$ and $\mathfrak{A}_1$ is isolated.

Theorem 4.6.4 (Grilliot [18]). *Let T be a consistent $\forall_2$ theory. Then T has an $\exists$-minimal pair.*

Using effective model theoretic forcing, we have the following effective analogue of this result.

Theorem 4.6.5. *Let T be a c.e., 2-complete $\forall_2$ theory, with $\mathcal{P}$ a model theoretic property effectively enforceable over T. Then there is an $\exists$-minimal pair of computable structures $\mathfrak{A}_0$, $\mathfrak{A}_1$, each with property $\mathcal{P}$.*

Proof. We can assume without loss of generality that property $\mathcal{P}$ includes decidability for atomic sentences of $\mathcal{L}(W)$; it is does not, we just combine this property with $\mathcal{P}$ using Theorem 3.3.9. We will construct two winning computable plays $\langle p^i_s \rangle$, with $i = 0, 1$, for the game $G(\mathcal{P})$ such that the compiled structures $\overline{\mathfrak{A}}_i = \overline{\mathfrak{A}}(p^i_\omega)$ form an $\exists$-minimal pair. This will be an $\exists$-minimal pair of computable structures since every element $a \in \overline{\mathfrak{A}}_i$ is of the form $a = c^{\overline{\mathfrak{A}}_i}$ for some $c \in W$.

In our plays $\langle p^i_s \rangle$, we will use the winning strategy for player$_\exists$ at the even stages while ensuring that $\overline{\mathfrak{A}}_i$ form an $\exists$-minimal pair at the odd stages. In order to do this, we need to meet the following requirements for all $\overline{a}_1$, $\overline{a}_0 \in W$ with $|\overline{a}_0| = |\overline{a}_0|$.

$(R_{\langle \overline{a}_0, \overline{a}_1 \rangle})$ If $\exists\text{-type}_{\overline{\mathfrak{A}}}(\overline{a}_0) = \exists\text{-type}_{\overline{\mathfrak{A}}}(\overline{a}_1) = \Gamma(\overline{x})$, then Γ has $\exists$ support.

As any $\exists$ type with $\exists$ support is maximal and isolated, this is enough.

The moves of player$_\forall$ will be structured as a priority argument. We arrange the requirements $(R_{\langle \overline{a}_0, \overline{a}_1 \rangle})$ by priority according to the standard ordering on the pairs $\langle \overline{a}_0, \overline{a}_1 \rangle$. For each stage s of the play and $\overline{a} \in W$, we define $\delta^i_s(\overline{a}, \overline{c}) = \bigwedge p^i_s$. Then set

$$\rho^i_{\overline{a},s}(\overline{x}) \;=\; \exists \overline{y}\, \delta^i_s(\overline{x}, \overline{y}) \tag{4.38}$$

We say a pair $\overline{a}_0$, $\overline{a}_1$ is *incompatible at stage s* if

$$T \vdash \neg \exists \overline{x} \left(\rho^0_{\overline{a}_0,s}(\overline{x}) \wedge \rho^1_{\overline{a}_1,s}(\overline{x}) \right) \tag{4.39}$$

Otherwise, we say the pair is *compatible*. This can be determined effectively because T is c.e. and 2-complete.

At odd stages s of the construction, we will be extending the previous choices p^i_{s-1} to some $\widehat{p}^i \in N_T$. Define $\widehat{\rho}^i_{\overline{a}_i}$ for $\widehat{p}^i$ as in (4.38). We say that a choice of extensions $\widehat{p}^i$ *challenges* the pair $\overline{a}_0$, $\overline{a}_1$ at stage s if $\overline{a}_0$, $\overline{a}_1$ are compatible at stage $s-1$ and

$$T \nvdash \forall \overline{x} \left(\rho^i_{\overline{a}_i,s-1}(\overline{x}) \to \widehat{\rho}^i_{\overline{a}_i}(\overline{x})\right) \tag{4.40}$$

for some $i = 0, 1$. Again, (4.40) can be determined effectively because T is c.e. and 2-complete. The following lemma shows the significance of (4.40).

Lemma 4.6.6. *If a choice of extensions $\widehat{p}^i$ challenges a pair $\overline{a}_0$, $\overline{a}_1$ at stage s, then we can choose some pair of extensions $p^i_s \supseteq p^i_{s-1}$ which make $\overline{a}_0$, $\overline{a}_1$ incompatible at stage s.*

Proof. If the pair $\widehat{p}^i$ already make the $\overline{a}_0$, $\overline{a}_1$ incompatible, there is nothing to show. Otherwise, there is some $i = 0, 1$ such that (4.40) holds. Assume without loss of generality that $i = 0$.

Let $\mathfrak{A}(\widehat{p}^0)$ be the canonical model for $\widehat{p}^0$. Then

$$\widehat{\rho}^0_{\overline{a}_0}(\overline{x}) \in \exists\text{-type}_{\mathfrak{A}(\widehat{p}^0)}(\overline{a}) \tag{4.41}$$

So, by (4.40), $\rho^0_{\overline{a}_i,s-1}$ does not support $\exists$-type$(\overline{a})$. As T is 2-complete, we know from Proposition 2.4.13 that a maximal $\exists$ type over T is isolated only if it has $\exists$ support. So there are at least two different maximal $\exists$ types containing $\rho^0_{\overline{a}_i,s-1}$.

Let Γ_1 be a maximal $\exists$ type containing $\rho^1_{\overline{a}_1,s-1}$. We use the fact that we can extend $\rho^0_{\overline{a}_0,s-1}$ to multiple maximal $\exists$ types to choose some Γ_0 which is different from Γ_1. Then there is some $\exists$ formula $\varphi(\overline{x})$ such that

$$\varphi(\overline{x}) \in \Gamma_0(\overline{x}) \quad \Leftrightarrow \quad \varphi(\overline{x}) \notin \Gamma_1(\overline{x}) \tag{4.42}$$

Assume without loss of generality that $\varphi \in \Gamma_0$. As $\varphi \notin \Gamma_1$, a maximal $\exists$ type, there is some $\exists$ formula $\psi \in \Gamma_1$ such that

$$T \vdash \forall \overline{x} \left(\psi(\overline{x}) \to \neg\varphi(\overline{x})\right) \tag{4.43}$$

As the pair $\widehat{p}^i$ extend the p^i_{s-1}, we have that $\varphi(\overline{a}_0)$ and $\psi(\overline{a}_0)$ are consistent with p^0_{s-1} and p^1_{s-1}, respectively, by the definition of $\rho^i_{\overline{a}_i,s-1}$. We use Proposition 4.2.4 to extend the p^i_{s-1} accordingly. ◻

Given this result, we can now proceed with our construction.

Stage $s = 2e$:
At this stage, we play the strategy for property $\mathcal{P}$.

Stage $s = 2e+1$:
If the moves p^i_{s-1} do not challenge any pair $\overline{a}_0$, $\overline{a}_1$ less than s, we set $p^i_s = p^i_{s-1}$. Otherwise, we let $\langle \overline{a}_0, \overline{a}_1 \rangle$ be the challenged pair of highest priority. From the argument above, we know that there are extensions $q^i \supseteq p^i_{s-1}$ in N_T making the pair $\overline{a}_0$, $\overline{a}_1$ incompatible. We let p^i_s be the first pair that we find.

It is clear that our structures $\mathfrak{A}_0$, $\mathfrak{A}_1$ have the property $\mathcal{P}$ because we applied our effective strategy at the even stages. We need only show that each requirement $(R_{\langle \overline{a}_0, \overline{a}_1 \rangle})$ is eventually met.

Take any pair $\overline{a}_i$ and let s_0 be the stage such that all requirements of higher priority are never challenged again. If this pair is challenged at some stage $t \geq s_0$, then p^i_t make the pair $\overline{a}_i$ incompatible and so $\exists$-type$(\overline{a}_0) \neq \exists$-type$(\overline{a}_1)$. Furthermore, this pair will never be challenged again after stage t.

Suppose this pair is never challenged after stage s_0. For all odd stages $s > s_0$, $p^i_s = p^i_{s-1}$ and hence trivially

$$T \vdash \forall \overline{x} \left(\rho^i_{\overline{a}_i, s}(\overline{x}) \rightarrow \rho^i_{\overline{a}_i, s+1}(\overline{x}) \right) \tag{4.44}$$

for $i = 0$, 1. At even stages $s > s_0$, p^i_s does not challenge the pair $\overline{a}_i$, so (4.44) holds at those stages by definition. By induction, for all $s > s_0$ and for $i = 0$, 1,

$$T \vdash \forall \overline{x} \left(\rho^i_{\overline{a}_i, s_0}(\overline{x}) \rightarrow \rho^i_{\overline{a}_i, s}(\overline{x}) \right) \tag{4.45}$$

It is clear that (4.45) implies that $\rho^i_{\overline{a}_i, s_0}$ is the $\exists$ support of $\exists$-type$(\overline{a}_i)$. $\square$

4.7 Directions for further research

The results of the previous two sections suggest that we should be able to use effective model theoretic forcing, or similar techniques, to construct computable models of $\forall_2$ theories with other additional properties. In particular, another type of model that we mentioned in Chapter 2 was an existentially saturated model. Millar has proven a characterization similar to Theorem 4.5.1 for decidable saturated models.

Theorem 4.7.1 (Millar [36]). *A theory T has a decidable saturated model if and only if the complete types of T are uniformly computable.*

The proof of this theorem is a modification of Harrington's proof of Theorem 4.5.1 where all types are realized instead of just the principal ones. Therefore, just as we adapted Harrington's proof to construct a computable, e.c., algebraically prime model, it seems that we should be able to adapt Millar's proof to construct a computable existentially saturated model. In particular, the relationship between existentially saturated models and saturated models suggests the following result.

Conjecture 4.7.2. If T is an $\forall_2$ theory such that the maximal $\exists$ types of T are uniformly computable, then T has a 1-decidable existentially saturated model.

Another interesting area for further research is the existence of e.c. extensions of models. Theorem 2.2.2 does not just give an e.c. model of an $\forall_2$ theory, it constructs this model as an extension of an existing model. We would like to know what conditions are sufficient for us to be able to do this effectively for any computable model of our theory.

While we do have such a result, it only covers a very small class of $\forall_2$ theories.

Definition 4.7.3. A theory T has the *amalgamation property* if, for any models $\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C} \models T$ with embeddings $e : \mathfrak{A} \to \mathfrak{B}$ and $f : \mathfrak{A} \to \mathfrak{C}$, there is a model $\mathfrak{D} \models T$ and embeddings $g : \mathfrak{B} \to \mathfrak{D}$ and $h : \mathfrak{C} \to \mathfrak{D}$ such that the diagram in Figure 4.1 commutes.

$$\begin{tikzcd}
& \mathfrak{B} \arrow[dr, dashed, "g"] & \\
\mathfrak{A} \arrow[ur, "e"] \arrow[dr, "f"'] & & \mathfrak{D} \\
& \mathfrak{C} \arrow[ur, dashed, "h"'] &
\end{tikzcd}$$

Figure 4.1: The amalgamation property

Theorem 4.7.4. *Let $\mathcal{L}$ be a language with finitely many symbols and no functions. Let T be a consistent, $\forall$-decidable $\forall_2$ theory such that $T_\forall$ has the amalgamation property. Finally, let $\mathfrak{B} \models T_\forall$ be computable. Then there is a computable e.c. model $\overline{\mathfrak{A}} \models T$ such that there is an effective embedding $f : \mathfrak{B} \hookrightarrow \overline{\mathfrak{A}}$.*

Proof. All we need to do is to show that, if we play the odd stages as player$_\forall$, we can construct a play p_ω and a computable function $f : \mathfrak{B} \to \overline{\mathfrak{A}}$ such that f is an embedding. The moves of player$_\exists$ will ensure that $\mathfrak{B}$ is computable and an e.c. model of T.

For each finite sequence $\bar{b} \in \mathfrak{B}$, let $\langle \bar{b} \rangle_{\mathfrak{B}}$ be the least substructure of $\mathfrak{B}$ containing $\bar{b}$.

Let $\{ b_k \}_{k \in \omega}$ be a computable enumeration of the elements of $\mathfrak{B}$. As $\mathcal{L}$ has finitely many constant symbols, let $n \in \omega$ be such that for any constant symbol c of $\mathcal{L}$, $c^{\mathfrak{B}} = b_k$ for some $k \leq n$. Then, as $\mathcal{L}$ has no function symbols,

$$\mathfrak{B}_e = \{ b_0, b_1, \dots, b_{n+e} \} \tag{4.46}$$

is a finite substructure of $\mathfrak{B}$ for all $e \in \omega$. In particular, this means that $\mathfrak{B}_e \models T_\forall$.

Along with our model $\overline{\mathfrak{A}}$, we will build a one-one function $f : \mathfrak{B} \to W$ in stages $f = \bigcup_s f_s$ such that $f : \mathfrak{B} \to \overline{\mathfrak{A}}$ is an embedding. Initially, we define $f_0(b_k) = -1$ for all k. Our moves at the odd stages will be as follows.

Stage $s = 2e + 1$:

For each $b \in \mathfrak{B}_e$ such that $f_{e-1}(b) = -1$, we pick a distinct element $c \in W$ not appearing in p_{s-1} and set $f_e(b) = c$. We leave $f_e(b)$ unchanged for all other b. We define the set

$$B_e = \{ \xi(f_e(b_{i_0}), \dots, f_e(b_{i_k})) \mid \xi \text{ is a literal and } \mathfrak{B}_e \models \xi(b_{i_1}, \dots, b_{i_k}) \} \tag{4.47}$$

This is just the atomic diagram of $\mathfrak{B}_e$ with the elements $b \in \mathfrak{B}_e$ replaced by the constants $f_e(b)$.

As $\mathfrak{B}_e \models T_\forall$, this set is consistent with T. If $e = 0$ then we are at the initial stage and we just set $p_1 = B_e$. If $e > 0$, then by induction $B_{e-1} \subseteq p_{s-1}$. As p_{s-1} is consistent with T, there is some $\mathfrak{C} \models T$ such that $\mathfrak{C} \models p_{s-1}$. Furthermore, f_e defines an embedding of $\mathfrak{B}_{e-1}$ into $\mathfrak{C}$ as B_{e-1} is the diagram of $\mathfrak{B}_{e-1}$. As $\mathfrak{B}_{e-1} \subseteq \mathfrak{B}_e$, and $T_\forall$ has the amalgamation property, there is some model $\mathfrak{D} \models T_\forall$ completing the diagram in Figure 4.1. So, as $f_e(b_{n+e})$ is a constant not appearing in p_{s-1}, $p_{s-1} \cup B_e$ is consistent with T. We extend p_{s-1} accordingly.

It is clear that f is computable. As $B_e \subseteq p_\omega$ for all e, $f : \mathfrak{B} \to \overline{\mathfrak{A}}$ is an embedding. $\square$

This is an extremely unsatisfying result. While many $\forall_2$ theories T in algebra have the amalgamation property for $T_\forall$, they almost always have function symbols in their language. Therefore, if we want a useful version of Theorem 4.7.4, we need to find a way to include function symbols in our language.

Chapter 5

Completeness Results for Analytic Properties

5.1 Introduction

In Chapter 4, our characterizations were for determining when a theory had a computable model. Another type of characterization in computable model theory is one that establishes alternate criteria for a model theoretic property. One example of this is the property of strong homogeneity.

Definition 5.1.1. A countable structure $\mathfrak{A}$ is *strongly homogeneous* if, for all finite $\overline{a}$, $\overline{b} \in \mathfrak{A}$, if $(\mathfrak{A}, \overline{a}) \equiv (\mathfrak{A}, \overline{b})$ then there is an automorphism of $\mathfrak{A}$ taking $\overline{a}$ to $\overline{b}$.

This is a nice property for a structure to have, as it says that the automorphism orbits are determined by the complete types realized in the structure. However, it can be difficult to test whether or not a structure is strongly homogeneous. As we will see later in this chapter, every model theoretic property $\mathcal{P}$ has a natural notion of complexity, the complexity of determining whether or not a (computable) structure has property $\mathcal{P}$. Determining whether there is an automorphism of $\mathfrak{A}$ taking $\overline{a}$ to $\overline{b}$ is the Σ^1_1 predicate

$$\text{(5.1)}\quad \mathrm{AUT}_{\mathfrak{A}}(\overline{a}, \overline{b}) =_{\text{def}} \begin{array}{l} \exists f : \mathfrak{A} \to \mathfrak{A} \Big[\forall \xi \in \mathcal{L}^{\text{atm}}, \overline{c} \in \mathfrak{A} \big(\mathfrak{A} \models \xi(\overline{c}) \leftrightarrow \mathfrak{A} \models \xi(f(\overline{c})\big) \wedge \\ \qquad \big(f(\overline{a}) = \overline{b}\big) \wedge \forall d \in \mathfrak{A}\, \exists c \in \mathfrak{A} \big(f(c) = d\big) \wedge \\ \qquad \forall c, d \in \mathfrak{A} \big(f(c) = f(d) \to c = d\big)\Big] \end{array}$$

This suggests that the complexity of strong homogeneity is Σ^1_1.

In contrast to strong homogeneity, we also have the seemingly weaker property of homogeneity.

Definition 5.1.2. A countable structure $\mathfrak{A}$ is *homogeneous* if, for all finite $\overline{a}$, $\overline{b} \in \mathfrak{A}$, if $(\mathfrak{A}, \overline{a}) \equiv (\mathfrak{A}, \overline{b})$ and $c \in \mathfrak{A}$, there is some $d \in \mathfrak{A}$ with $(\mathfrak{A}, \overline{a}, c) \equiv (\mathfrak{A}, \overline{b}, d)$.

It is clear that any strongly homogeneous structure is also homogeneous; we just use the automorphism from $\overline{a}$ to $\overline{b}$ to pick d given c. Shelah [45] has shown that these two properties are, in fact, equivalent. This is a nice result because homogeneity is definable in $\mathcal{L}_{\omega_1,\omega}$ by the sentence

$$(5.2)\quad \varphi_{\mathrm{HOM}} =_{\mathrm{def}} \forall \overline{x}, \overline{y}, z\, \exists w \left[\Big(\bigwedge_{\psi \in \mathcal{L}} (\psi(\overline{x}) \leftrightarrow \psi(\overline{y})) \Big) \rightarrow \Big(\bigwedge_{\chi \in \mathcal{L}} (\chi(\overline{x}, z) \leftrightarrow \chi(\overline{y}, w)) \Big) \right]$$

In other words, for any structure $\mathfrak{A}$, $\mathfrak{A} \models \varphi_{\mathrm{HOM}}$ if and only if $\mathfrak{A}$ is homogeneous.

The formula $\bigwedge_{\psi \in \mathcal{L}} (\psi(\overline{x}) \leftrightarrow \psi(\overline{y}))$ is a computable Π_ω formula, and hence φ_{HOM} is a computable $\Pi_{\omega+2}$ sentence. Since homogeneity and strong homogeneity are equivalent, this means that their complexity is at most $\Pi^0_{\omega+2}$. Therefore, homogeneity is a useful characterization of strong homogeneity in computable structures.

If we can establish a lower bound on the complexity of a property $\mathcal{P}$, this means that any property $\mathcal{Q}$ with complexity less than this bound cannot be equivalent to $\mathcal{P}$. Such a result would be a "anticharacterization" of sorts; we can dismiss candidates for a characterization by simply looking at their algorithmic complexity. In this chapter, we are particularly interested in model theoretic properties whose lower bound is properly analytic. Since checking whether a computable infinitary sentence is satisfied by a computable model is hyperarithmetic, this would mean that, unlike homogeneity, there is no such sentence defining this property.

In this chapter, we will make more precise what we mean by the complexity of a model theoretic property, and show how to prove lower bounds for the complexity of a property. We will then use these techniques to prove anticharacterizations for several analytic properties.

5.2 Complexity of model theoretic properties

To establish lower bounds for the complexity of model theoretic properties, we first need to know how to establish them for predicates of ω. One way of doing this is through 1-reductions.

Definition 5.2.1. For any predicates $\mathcal{P}$, $\mathcal{Q}$ on ω, we say that $\mathcal{P}$ is 1-*reducible* to $\mathcal{Q}$, which we write $\mathcal{P} \leq_{1:1} \mathcal{Q}$ if there is some computable, one-one function $f : \omega \to \omega$ such that, for all $e \in \omega$, $\mathcal{P}(e)$ if and only if $\mathcal{Q}(f(e))$.

Soare [46] is an excellent survey of the properties of 1-reductions. One well-known property is that, whenever Γ is a class in the analytic or hyperarithmetic hierarchy, if $\mathcal{Q} \in \Gamma$ and $\mathcal{P} \leq_{1:1} \mathcal{Q}$, then $\mathcal{P} \in \Gamma$. This fact gives us the following useful definition for measuring lower bounds on algorithmic complexity.

Definition 5.2.2. Let Γ be a collection of predicates on ω. A predicate $\mathcal{P}$ on ω is Γ-*hard*, which we write $\Gamma \leq_{1:1} \mathcal{P}$, if $\mathcal{Q} \leq_{1:1} \mathcal{P}$ for all $\mathcal{Q} \in \Gamma$. We say that $\mathcal{P}$ is Γ-*complete*, which we write $\mathcal{P} \equiv_{1:1} \Gamma$ if $\mathcal{P}$ is Γ-hard and $\mathcal{P} \in \Gamma$.

Therefore, if we can uniquely associate a model theoretic property with a predicate on ω, then we not only have a way of assigning it a level of complexity, we also have a way to establish a lower bound for this complexity. As usual, let $\mathcal{L}$ be a computable language and let θ_n be an enumeration of the atomic sentences of $\mathcal{L}(\omega)$, the language where we add a constant for each element in ω. Then we can associate each $f \in 2^\omega$ with a set of atomic sentences.

Recall the definition of a =-closed set from Definition 3.2.2. As in Proposition 3.2.3, it is clear than any =-closed function corresponds to a structure $\mathfrak{M}_f$ on $\mathcal{L}$ whose diagram is given by

$$\theta_n \in \operatorname{diag}(\mathfrak{M}_f) \quad \Leftrightarrow \quad f(n) = 1 \tag{5.3}$$

This structure is computable whenever f is. Moreover, it is clear that any function f can be uniformly transformed to a =-closed function $\widehat{f}$ such that $f = \widehat{f}$ whenever f is =-consistent. So we can associate with each computable function $f \in 2^\omega$ a unique computable structure $\mathfrak{M}_f$.

Furthermore, this correspondence gives all of the computable structures on $\mathcal{L}$. For any computable structure $\mathfrak{M}$, we have that $\mathfrak{M} = \mathfrak{M}_{f_{\mathfrak{M}}}$, where

$$f_{\mathfrak{M}}(n) = 1 \quad \Leftrightarrow \quad \theta_n \in \operatorname{diag}(\mathfrak{M}_f) \tag{5.4}$$

This means that we can identify the computable structures of $\mathcal{L}$ with the computable functions in 2^ω. This provides us with a Π^0_2 enumeration of all the computable structures on $\mathcal{L}$. As this enumeration will be very useful in this and the following chapters, we give it a special name.

Definition 5.2.3. For each $e \in \omega$ such that $\{e\}$ is total, the e^{th} *computable structure of* $\mathcal{L}$, which we denote $\mathfrak{M}_e$, is the unique structure satisfying (5.3) for the function $\widehat{\{e\}}$.

Given this enumeration of the computable structures of $\mathcal{L}$, we can identify any model theoretic property $\mathcal{P}$ with the predicate

$$\mathrm{MOD}_{\mathcal{P}}(e) =_{\text{def}} \{e\} \text{ is total and } \mathfrak{M}_e \text{ has property } \mathcal{P} \tag{5.5}$$

The complexity of property $\mathcal{P}$ is then the complexity of this set. Note that, since the totality of $\{e\}$ is Π^0_2-complete, this definition means that any property has complexity at least Π^0_2. This is true even if we can decide, given that $\{e\}$ total, whether $\mathfrak{M}_e$ has property $\mathcal{P}$. However, as most of the properties we will examine have nonarithmetic lower bounds, this will not be a problem. Furthermore, from now on, when we write $\mathfrak{M}_e$, we will assume that $\{e\}$ is total.

A more serious problem is that this definition of complexity depends on the language $\mathcal{L}$. As we vary $\mathcal{L}$, we change the structure $\mathfrak{M}_e$ represented by each total function $\{e\}$. In some cases, this can change the complexity of a property. A simple example of this is the property of ultrahomogeneity.

Definition 5.2.4. A countable structure $\mathfrak{A}$ is *ultrahomogeneous* if, for all $\overline{a}, \overline{b} \in \mathfrak{A}$ such that $(\mathfrak{A}, \overline{a}) \equiv_0 (\mathfrak{A}, \overline{b})$ and all $c \in \mathfrak{A}$, there is a $d \in \mathfrak{A}$ such that $(\mathfrak{A}, \overline{a}, c) \equiv_0 (\mathfrak{A}, \overline{b}, d)$.

As with homogeneity, we can define ultrahomogeneity by a computable infinitary formula.

$$\varphi_{\mathrm{ULT}} =_{\mathrm{def}} \forall \overline{x}, \overline{y}, z\, \exists w \left[\Big(\bigwedge_{\mathcal{L}^{\mathrm{atm}}} (\psi(\overline{x}) \leftrightarrow \psi(\overline{y})) \Big) \to \Big(\bigwedge_{\mathcal{L}^{\mathrm{atm}}} (\psi(\overline{x}, z) \leftrightarrow \psi(\overline{y}, w)) \Big) \right] \tag{5.6}$$

However, the complexity of φ_{ULT} depends on $\mathcal{L}^{\mathrm{atm}}$. If $\mathcal{L}$ is a finite relational language, then there are only finitely many atomic formulae with free variables $\overline{x}$. In this case, φ_{ULT} is an $\forall_2$ first-order sentence, and hence ultrahomogeneity is, at most, Π^0_2.

However, as the following result shows, this is not true for a language with infinitely many relations.

Theorem 5.2.5. *Let $\mathcal{L}$ be the language with signature $\{ R, P_k \}_{k \in \omega}$, where R is a binary relation and each of the P_k is a unary predicate. Then ultrahomogeneity is Π^0_3-complete.*

Proof. Since this language has infinitely many relational symbols, for any sequence $\overline{x}$, there are infinitely many atomic formulae with free variables $\overline{x}$. Hence φ_{ULT} is no longer first-order. It is, however, equivalent to a computable Π_3 formula in $\mathcal{L}_{\omega_1,\omega}$. Therefore, ultrahomogeneity is at most Π^0_3 in this language.

So we need only show that ultrahomogeneity is Π^0_3-hard. Take any computable predicate $\mathcal{P}(x, y, z, n)$. We need to construct a uniformly computable sequence of structures $\mathfrak{A}_n$ such that

$$\mathfrak{A}_n \text{ is ultrahomogeneous} \quad \Leftrightarrow \quad \forall x\, \exists y\, \forall z\, \mathcal{P}(x, y, z, n) \tag{5.7}$$

Our reduction will then be a computable function whose range is the set of indices for this sequence. We can ensure that f is one-one by padding the algorithms for these structures $\mathfrak{A}_n$.

To construct this computable sequence, we first construct two simple structures $\mathcal{G}_\infty$ and $\mathcal{G}_{\mathrm{fin}}$. The idea is that the R-connected components of each $\mathfrak{A}_n$ will be isomorphic either to $\mathcal{G}_\infty$ or to $\mathcal{G}_{\mathrm{fin}}$. The structure $\mathcal{G}_\infty$ will have domain ω. It will be the complete irreflexive graph on ω with respect to R. That is, for all n, $m \in \omega$,

$$\mathcal{G}_\infty \vDash (n\ R\ m) \quad \Leftrightarrow \quad n \neq m \tag{5.8}$$

Furthermore, for each k, $\ell \in \omega$,

$$\mathcal{G}_\infty \vDash P_k(\ell) \quad \Leftrightarrow \quad \begin{cases} j \geq k \text{ and } \ell = 2\langle i, j\rangle + 1 \text{ for some } i \\ \ell \neq 0 \text{ is even} \end{cases} \tag{5.9}$$

We can think of $\mathcal{G}_\infty$ as an equivalence class where all elements $a \in \mathcal{G}_\infty$ are labeled by some element $n_a \in \omega \cup \{\, \infty \,\}$, where

$$\mathcal{G}_\infty \models P_k(a) \quad \Leftrightarrow \quad k < n_a \tag{5.10}$$

$\mathcal{G}_\infty$ has a unique element a labeled by 0, namely $a = 0$. For each $\ell \in \omega$ with $\ell \geq 1$, $\mathcal{G}_\infty$ has infinitely many elements b_i^ℓ labeled by ℓ, namely $b_i^\ell = \langle i, \ell - 1 \rangle$. Finally, $\mathcal{G}_\infty$ has infinitely many elements b_i^∞ labeled by ∞, namely $b_i^\infty = 2(i+1)$. This concept is illustrated in Figure 5.1.

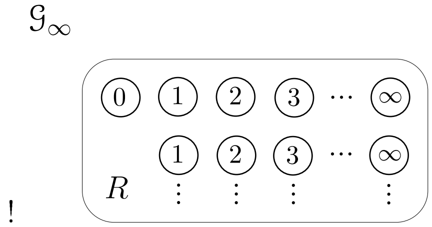

Figure 5.1: The structure $\mathcal{G}_\infty$

Because $\mathcal{G}_\infty$ is the complete irreflexive graph on ω with respect to R, any permutation $\pi : \omega \to \omega$ such that for all k, $a \in \omega$

$$\mathcal{G}_\infty \models P_k(a) \quad \Leftrightarrow \quad \mathcal{G}_\infty \models P_k(\pi(a)) \tag{5.11}$$

is an automorphism. Therefore, $\mathcal{G}_\infty$ has the following property.

Lemma 5.2.6. *Let $\mathcal{G}$ and $\mathcal{G}'$ be two copies of $\mathcal{G}_\infty$. For any $\overline{a} \in \mathcal{G}$, $\overline{b} \in \mathcal{G}'$*

$$(\mathcal{G}, \overline{a}) \equiv_0 (\mathcal{G}', \overline{b}) \quad \Leftrightarrow \quad (\mathcal{G}, \overline{a}) \cong (\mathcal{G}', \overline{b}) \tag{5.12}$$

Proof. The only nontrivial direction is $(\Rightarrow)$. Assume that $(\mathcal{G}, \overline{a}) \equiv_0 (\mathcal{G}', \overline{b})$. To show that $(\mathcal{G}, \overline{a}) \cong (\mathcal{G}', \overline{b})$, it is enough to show that the corresponding elements of $\overline{a}$ and $\overline{b}$ agree on the predicates P_k. That is, if $n = |\overline{a}| = |\overline{b}|$, we need only show that, for all $i \leq n$ and $k \in \omega$,

$$\mathcal{G} \models P_k(a_i) \quad \Leftrightarrow \quad \mathcal{G}' \models P_k(b_i) \tag{5.13}$$

This fact is immediate from $(\mathcal{G}, \overline{a}) \equiv_0 (\mathcal{G}', \overline{b})$. ∎

The structure $\mathcal{G}_{\text{fin}}$ is exactly like $\mathcal{G}_\infty$ except that no nodes are labeled by ∞. That is, there are no elements $a \in \mathcal{G}_{\text{fin}}$ such that $\mathcal{G}_{\text{fin}} \models P_k(a)$ for all $k \in \omega$. For convenience, we can consider $\mathcal{G}_{\text{fin}}$ as the substructure of $\mathcal{G}_\infty$ with domain

$$\operatorname{dom}(\mathcal{G}_{\text{fin}}) = \{\, 0 \,\} \cup \{\, 2m + 1 \,\}_{m \in \omega} \tag{5.14}$$

Lemma 5.2.7. *For all $\overline{a} \in \mathcal{G}_{\text{fin}}$ there is some $\overline{b} \in \mathcal{G}_\infty$ such that $(\mathcal{G}_{\text{fin}}, \overline{a}) \equiv_0 (\mathcal{G}_\infty, \overline{b})$.*

Proof. The structure $\mathcal{G}_{\text{fin}}$ is embeddable in $\mathcal{G}_\infty$ in the obvious way. Let $\overline{b}$ be the image of $\overline{a}$ under this embedding. Then, for atomic formulae ξ

$$\mathcal{G}_{\text{fin}} \models \xi(\overline{a}) \quad \Leftrightarrow \quad \mathcal{G}_\infty \models \xi(\overline{b}) \tag{5.15}$$

This gives the desired result. ■

Our structures $\mathfrak{A}_n$ will be computable disjoint unions of copies of the structures $\mathcal{G}_{\text{fin}}$ and $\mathcal{G}_\infty$. For each $x \in \omega$, we construct a computable structure $\mathfrak{B}^x_n$. As with $\mathcal{G}_\infty$, $\mathfrak{B}^x_n$ is the complete irreflexive graph on ω with respect to R. For each $k, \ell \in \omega$, we define

$$\mathfrak{B}^x_n \models P_k(\ell) \quad \Leftrightarrow \quad \begin{cases} k = 0 \text{ or } \ell = 0 \\ \bigwedge_{z \le k} \mathcal{P}(x, j, z, n), \text{ where } \ell = 2\langle i, j\rangle + 2 \\ j \ge k, \text{ where } \ell = 2\langle i, j\rangle + 1 \end{cases} \tag{5.16}$$

Finally, let $\mathfrak{B}^\infty_n \cong \mathcal{G}_\infty$. For each n, let $\mathfrak{A}_n$ be the computable disjoint union of $\mathfrak{B}^x_n$ for each $x \in \omega \cup \{\infty\}$. Then $\mathfrak{A}_n$ has the following property.

Lemma 5.2.8. *$\mathfrak{A}_n$ is ultrahomogeneous if and only if $\mathfrak{B}^x_n \cong \mathcal{G}_\infty$ for all $x \in \omega$.*

Proof. ($\Rightarrow$) Suppose for a contradiction that there is some $x \in \omega$ such that $\mathfrak{B}^x_n \cong \mathcal{G}_{\text{fin}}$. Take any element $a \in \mathfrak{B}^x_n$. As $\mathfrak{B}^x_n$ is a substructure of $\mathfrak{A}_n$, $(\mathfrak{B}^x_n, a) \equiv_0 (\mathfrak{A}_n, a')$.

By Lemma 5.2.7, there is some $b \in \mathcal{G}_\infty$ such that $(\mathfrak{B}^x_n, a) \equiv_0 (\mathcal{G}_\infty, b)$. By construction, $\mathfrak{B}^\infty_n$ is a copy of $\mathcal{G}_\infty$. Let $b' \in \mathfrak{A}_n$ be the image of b under the natural embedding. Then $(\mathcal{G}_\infty, b) \equiv_0 (\mathfrak{A}_n, b')$. Hence $(\mathfrak{A}_n, a) \equiv_0 (\mathfrak{A}_n, b')$.

Now let $c \in \mathfrak{B}^\infty_n$ be an element which is labeled by ∞ (i.e. $\mathfrak{A}_n \models P_k(c)$ for all $k \in \omega$). As $\mathfrak{A}_n$ is ultrahomogeneous, there is some $d \in \mathfrak{A}_n$ such that $(\mathfrak{A}_n, a, d) \equiv_0 (\mathfrak{A}_n, b', c)$. As $\mathfrak{A}_n \models (b' \; R \; c)$, $\mathfrak{A}_n \models (a \; R \; d)$. So $d \in \mathfrak{B}^x_n$. Furthermore, $\mathfrak{A}_n \models P_k(d)$ for all $k \in \omega$, and hence $\mathfrak{B}^x_n \models P_k(d)$ for all $k \in \omega$. This contradicts the claim that $\mathfrak{B}^x_n \cong \mathcal{G}_{\text{fin}}$.

($\Leftarrow$) Take any $\overline{a}$, $\overline{b} \in \mathfrak{A}_n$ such that $(\mathfrak{A}_n, \overline{a}) \equiv_0 (\mathfrak{A}_n, \overline{b})$. For each $x \in \omega \cup \{\infty\}$, let $\overline{a}_x$ be exactly those elements of $\overline{a}$ that are in $\mathfrak{B}^x_n$. Let $\overline{b}_x$ be the corresponding elements in $\overline{b}$ (i.e. $\overline{b}_x = b_{i_1} \cdots b_{i_k}$ if $\overline{a}_x = a_{i_1} \cdots a_{i_k}$). For all b_i, $b_j \in \overline{b}$, b_i and b_j are in the same copy of $\mathcal{G}_\infty$ if and only if $\mathfrak{A}_n \models (b_i \; R \; b_j)$ since $\mathcal{G}_\infty$ is a complete graph with respect to R. So the elements $\overline{b}_x$ are each contained in a distinct copy of $\mathcal{G}_\infty$. Let $\pi : \omega \cup \{\infty\} \to \omega \cup \{\infty\}$ be a permutation such that $\overline{b}_x \in \mathfrak{B}^{\pi(x)}_n$ for all x.

We know from Lemma 5.2.6 that $(\mathfrak{B}^x_n, \overline{a}_x) \cong (\mathfrak{B}^{\pi(x)}_n, \overline{b}_x)$ for all x. As the $\mathfrak{B}^x_n$ are disjoint, we can extend these individual isomorphisms to an automorphism of $\mathfrak{A}_n$ and hence $(\mathfrak{A}_n, \overline{a}) \cong (\mathfrak{A}_n.\overline{b})$. Now take any $c \in \mathfrak{A}_n$. Let $d \in \mathfrak{A}_n$ be the image of c under the automorphism taking $\overline{a}$ to $\overline{b}$. Then $(\mathfrak{A}_n, \overline{a}, d) \cong (\mathfrak{A}_n.\overline{b}, d)$ and hence $(\mathfrak{A}_n, \overline{a}, d) \equiv_0 (\mathfrak{A}_n, \overline{b}, d)$. ■

It is clear from (5.16) that, for $x \in \omega$, $\mathfrak{B}^x_n \cong \mathcal{G}_\infty$ whenever $\exists y \, \forall z \, \mathcal{P}(x, y, z, n)$; otherwise $\mathfrak{B}^x_n \cong \mathcal{G}_{\text{fin}}$. Hence our claim follows from Lemma 5.2.8. □

One solution to this problem is to restrict ourselves to languages with a finite signature. However, most languages in algebra have function symbols. As the following result shows, even a single binary function can effect the complexity of ultrahomogeneity.

Theorem 5.2.9. *Let $\mathcal{L}'$ be the language with signature $\{ f \}$, where f is a 2-place function. Then* $\mathrm{ULT}_{\mathcal{L}'} \equiv_{1:1} \Pi^0_3$.

Proof. Our proof will be roughly the same as Theorem 5.2.5 provided that we can construct analogues of the structures $\mathcal{G}_\infty$ and $\mathcal{G}_{\text{fin}}$ in this language. As before, the structure $\widehat{\mathcal{G}}_\infty$ will have domain ω. For all $n, m \in \omega$, we let

$$f^{\widehat{\mathcal{G}}_\infty}(n,m) = \begin{cases} 0 & n = m \in \{ 0, \langle i,0\rangle + 1 \}_{i\in\omega} \\ \langle i,0\rangle + 1 & n = m = \langle i,j\rangle + 1,\ j \geq 1 \\ \langle i,j+1\rangle + 1 & n = 0,\ m = \langle i,j\rangle + 1,\ \mathcal{G}_\infty \vDash P_j(i) \\ \langle i,j+1\rangle + 1 & n = \langle i,j\rangle + 1,\ m = 0 \\ n & \text{otherwise} \end{cases} \tag{5.17}$$

The idea behind this definition is that f codes the structure illustrated in Figure 5.1. For each $a \in \mathcal{G}_\infty$, we define the function

$$g_a(x) = f^{\widehat{\mathcal{G}}_\infty}\Big(\underbrace{f^{\widehat{\mathcal{G}}_\infty}\big(\underbrace{f^{\widehat{\mathcal{G}}_\infty}(a,a)}_{0 \text{ or } \langle i,0\rangle+1}, f^{\widehat{\mathcal{G}}_\infty}(a,a)\big)}_{0}, x\Big) \tag{5.18}$$

Then we label the element a with the least $n_a \in \omega \cup \{ \infty \}$ such that

$$g_a^{n_a}(f^{\widehat{\mathcal{G}}_\infty}(a,a)) = g_a^{n_a+1}(f^{\widehat{\mathcal{G}}_\infty}(a,a)) \tag{5.19}$$

As in Theorem 5.2.5, $\widehat{\mathcal{G}}_\infty$ has the following property.

Lemma 5.2.10. *Let $\mathcal{G}$ and $\mathcal{G}'$ be two copies of $\widehat{\mathcal{G}}_\infty$. For any $\overline{a} \in \mathcal{G}$, $\overline{b} \in \mathcal{G}'$*

$$(\mathcal{G}, \overline{a}) \equiv_0 (\mathcal{G}', \overline{b}) \quad \Leftrightarrow \quad (\mathcal{G}, \overline{a}) \cong (\mathcal{G}', \overline{b}) \tag{5.20}$$

Proof. The only nontrivial direction is ($\Rightarrow$). Assume that $(\mathcal{G}, \overline{a}) \equiv_0 (\mathcal{G}', \overline{b})$. We define a copy $\overline{\mathcal{G}}$ of $\mathcal{G}_\infty$ as follows. Let $\pi : \widehat{\mathcal{G}}_\infty \to \mathcal{G}$ be an isomorphism. The domain of $\overline{\mathcal{G}}$ is the set

$$\mathrm{dom}(\overline{\mathcal{G}}) = \{ \pi(a) \mid a = 0 \text{ or } a = \langle i,0\rangle + 1, i \in \omega \} \tag{5.21}$$

Furthermore, for all $\ell \in \overline{\mathcal{G}}$, we define

$$\overline{\mathcal{G}} \vDash P_k(\ell) \quad \Leftrightarrow \quad \pi^{-1}(\ell) = \langle i,0\rangle + 1 \text{ and } f^{\widehat{\mathcal{G}}_\infty}(0, \langle i,k\rangle + 1) \neq 0 \tag{5.22}$$

It is clear from (5.17) that this defines a copy of $\mathcal{G}_\infty$ with $\pi(0)$ being the unique element $x \in \overline{\mathcal{G}}$ such that $\overline{\mathcal{G}} \nvDash P_k(x)$ for all $k \in \omega$. Similarly, we define $\overline{\mathcal{G}'}$ from an isomorphism $\pi' : \widehat{\mathcal{G}}_\infty \to \mathcal{G}'$.

Let $\overline{c} \in \overline{\mathcal{G}}$ be the elements

$$c_i = \begin{cases} a_i & \pi^{-1}(a_i) = 0 \\ \pi(\langle i, 0\rangle + 1) & \pi^{-1}(a_i) = \langle i, j\rangle + 1 \end{cases} \tag{5.23}$$

Similarly, we define $\overline{d} \in \overline{\mathcal{G}'}$ from $\overline{b}$. As $(\mathcal{G}, \overline{a}) \equiv_0 (\mathcal{G}', \overline{b})$, it follows from (5.22) that $(\overline{\mathcal{G}}, \overline{c}) \equiv_0 (\overline{\mathcal{G}'}, \overline{d})$. So $(\overline{\mathcal{G}}, \overline{c}) \cong (\overline{\mathcal{G}'}, \overline{d})$ by Lemma 5.2.6.

Let $\rho : \overline{\mathcal{G}} \to \overline{\mathcal{G}'}$ be such an isomorphism. Note that $\rho(\pi(0)) = \rho(\pi'(0))$ by our construction of $\overline{\mathcal{G}}$ and $\overline{\mathcal{G}'}$. Hence for all $i \in \omega$, there is a unique $k \in \omega$ such that

$$\rho(\pi(\langle i, 0\rangle + 1)) = \pi'(\langle k, 0\rangle + 1) \tag{5.24}$$

Furthermore, for all $k \in \omega$, there is a unique $i \in \omega$ such that (5.24) holds. Therefore, we can extend ρ to a bijection $\widehat{\rho} : \mathcal{G} \to \mathcal{G}'$ by defining

$$\widehat{\rho}(a) = \begin{cases} \rho(a) & a \in \overline{\mathcal{G}} \\ \pi'(\langle k, j+1\rangle + 1) & \begin{array}{l} \pi^{-1}(a) = \langle i, j+1\rangle + 1 \text{ and} \\ \rho(\pi(\langle i, 0\rangle + 1)) = \pi'(\langle k, 0\rangle + 1) \end{array} \end{cases} \tag{5.25}$$

We need only show that $\widehat{\rho}$ is an isomorphism.

Take any a, $b \in \mathcal{G}$. We need to show that

$$\widehat{\rho}(f^{\mathcal{G}}(a, b)) = f^{\mathcal{G}'}(\widehat{\rho}(a), \widehat{\rho}(b)) \tag{5.26}$$

We have five cases to consider.

Case $a = b = \pi(0)$: This case is immediate from the fact that $f^{\mathcal{G}}(\pi(0), \pi(0)) = \pi(0)$ and $f^{\mathcal{G}'}(\pi'(0), \pi'(0)) = \pi'(0)$.

Case $a = b = \pi(\langle i, j\rangle + 1)$: Let $k \in \omega$ be such that $\rho(\pi(\langle i, 0\rangle + 1) = \pi'(\langle k, 0\rangle + 1)$. By (5.17), we have that

$$\begin{aligned} \widehat{\rho}(f^{\mathcal{G}}(a, b)) &= \widehat{\rho}(\pi(\langle i, 0\rangle + 1)) = \pi'(\langle k, 0\rangle + 1) \\ &= f^{\mathcal{G}'}(\pi'(\langle k, j\rangle + 1), \pi'(\langle k, j\rangle + 1)) \end{aligned} \tag{5.27}$$

So (5.26) follows from the definition of $\widehat{\rho}$.

Case $a = \pi(0)$**,** $b = \pi(\langle i, j\rangle + 1)$: Let $k \in \omega$ be such that $\rho(\langle i, 0\rangle + 1) = \pi'(\langle k, 0\rangle + 1)$. Since $\rho : \overline{\mathcal{G}} \to \overline{\mathcal{G}'}$ is an isomorphism, it follows from (5.17) that

$$\begin{aligned} \widehat{\rho}(f^{\mathcal{G}}(a, b)) &= \begin{cases} \widehat{\rho}(\pi(\langle i, j+1\rangle + 1)) & \text{if } \mathcal{G}_\infty \vDash P_j(i) \\ \widehat{\rho}(\pi(0)) & \text{otherwise} \end{cases} \\ &= \begin{cases} \pi'(\langle k, j+1\rangle + 1)) & \text{if } \mathcal{G}_\infty \vDash P_j(k) \\ \widehat{\rho}(\pi'(0)) & \text{otherwise} \end{cases} \\ &= f^{\mathcal{G}'}(\widehat{\rho}(a), \widehat{\rho}(b)) \end{aligned} \tag{5.28}$$

Case $a = \pi(\langle i, j\rangle + 1)$**,** $b = \pi(0)$: Let $k \in \omega$ be such that $\rho(\pi(\langle i, 0\rangle + 1) = \pi'(\langle k, 0\rangle + 1)$. By (5.17), we have that

$$\begin{aligned} \widehat{\rho}(f^{\mathcal{G}}(a,b)) &= \widehat{\rho}(\pi(\langle i, j+1\rangle + 1)) = \pi'(\langle k, j+1\rangle + 1) \\ &= f^{\mathcal{G}'}(\pi'(\langle k, j\rangle + 1), \pi'(0)) \end{aligned} \tag{5.29}$$

So (5.26) follows from the definition of $\widehat{\rho}$.

Case $a \neq b$ **and** a**,** $b \neq \pi(0)$: By construction $\widehat{\rho}(a) \neq \widehat{\rho}(b)$ and $\widehat{\rho}(a)$, $\widehat{\rho}(b) \neq \pi'(0)$. So by (5.17), we have that

$$\widehat{\rho}(f^{\mathcal{G}}(a,b)) = \widehat{\rho}(a) = f^{\mathcal{G}'}(\widehat{\rho}(a), \widehat{\rho}(b)) \tag{5.30}$$

■

We define $\widehat{\mathcal{G}}_{\text{fin}}$ similarly. As with Theorem 5.2.5, we can consider $\widehat{\mathcal{G}}_{\text{fin}}$ as the substructure of $\widehat{\mathcal{G}}_{\infty}$ with the domain given in (5.14). It is clear from (5.17) that this domain is closed under f.

Now let $\mathfrak{B}^x_n$ be as in Theorem 5.2.5. By replacing $\mathcal{G}_\infty$ with $\mathfrak{B}^x_n$ in (5.17), we can define the structure $\widehat{\mathfrak{B}}^x_n$. Again, each $\widehat{\mathfrak{B}}^x_n$ is isomorphic to either $\widehat{\mathcal{G}}_\infty$ or to $\widehat{\mathcal{G}}_{\text{fin}}$.

For each n we let the domain of $\widehat{\mathfrak{A}}_n$ be the disjoint union of $\widehat{\mathfrak{B}}^x_n$ for $x \in \omega \cup \{\infty\}$. We keep f as defined on each of the $\widehat{\mathfrak{B}}^x_n$, and for all a, $b \in \widehat{\mathfrak{A}}_n$ that are not in the same copy of $\widehat{\mathfrak{B}}^x_n$, we define $f^{\widehat{\mathfrak{A}}_n}(a,b) = b$. Then it is clear from (5.17) that two elements a, $b \in \widehat{\mathfrak{A}}_n$ are in the same copy of $\widehat{\mathfrak{B}}^x_n$ if and only if

$$a \neq b \quad \text{and} \quad f^{\widehat{\mathfrak{A}}_n}(a,b) \neq b \tag{5.31}$$

This means that for any $\overline{a}$, $\overline{b} \in \widehat{\mathfrak{A}}_n$ such that $(\widehat{\mathfrak{A}}_n, \overline{a}) \equiv_0 (\widehat{\mathfrak{A}}_n, \overline{b})$, we can partition these elements exactly as we did in Lemma 5.2.8. So our proof is the same as Theorem 5.2.5, given Lemma 5.2.10. □

So, as Theorem 5.2.5 and Theorem 5.2.9 illustrate, the complexity of a model theoretic property depends upon the choice of signature. Our solution to this problem will be to restrict ourselves to the language with a single binary relation R. This is because most interesting languages contain either at least one binary relation, or at least one binary function which can act as a binary relation. Therefore, if we can show that a property has complexity Γ in any computable language, and that it is Γ-complete in the language with relation R, this will be sufficient for our analysis. Therefore, unless otherwise noted, the complexity of a model theoretic property will be defined relative to this language.

5.3 Saturation

The first analytic property of interest to us is saturation, which we defined in Definition 2.3.5. First, we need determine an upper bound for the complexity of saturation. If we let ϕ_n be an enumeration of the formulae $\varphi(x)$ of $\mathcal{L}(\overline{a})$, for $\overline{a} \in \omega$, then we can associate any function $f \in 2^\omega$ with a set of such formulae. For convenience, we write $\varphi \in f$ if $\varphi = \phi_n$ and $f(n) = 1$. Then whether or not f defines a complete type of $\mathrm{Th}(\mathfrak{M}_e, \overline{a})$ is the predicate

$$\text{(5.32)} \qquad \mathrm{TYPE}(f, e) =_{\mathrm{def}} \forall n \left(\big(\mathfrak{M}_e \models \exists x \bigwedge_{f(k)=1, k \leq n} \phi_k(x)\big) \wedge (\phi_n \in f \wedge \neg\phi_n \in f) \right)$$

As any element of $\mathcal{L}(\overline{a})$ is $\forall_n$ for some n, this is a Π^0_ω predicate. Similarly, whether or not the type determined by f is realized in $\mathfrak{M}_e$ is the predicate

$$\text{(5.33)} \qquad \mathrm{REAL}(f, e) =_{\mathrm{def}} \mathrm{TYPE}(f, e) \wedge \mathfrak{M}_e \models \exists x \bigwedge_{f(n)=1} \phi_n(x)$$

This predicate is $\Sigma^0_{\omega+1}$. As the complexity of saturation is that of the predicate $\forall f\, \mathrm{REAL}(f, e)$, this shows that saturation is at most Π^1_1 over any language.

Therefore, as we said in section 5.2, we want to show that saturation is Π^1_1-complete in the language with a single binary relation R. However, in general it is difficult to specify all of the complete types of a structure in this language. It is much easier to do this in a language with only unary predicates, as demonstrated in the following result.

Theorem 5.3.1. *Let $\mathcal{L}$ be the language with countably many unary predicates P_n. Then saturation is Π^1_1-complete in this language.*

Proof. Instead of reducing an arbitrary Π^1_1 predicate to saturation, we will reduce a particular Π^1_1-complete predicate. As 1-reducibility is transitive, this will be enough.

Definition 5.3.2. A *tree* T is a subset of $\omega^{<\omega}$ that is closed under initial segments. That is, for any σ, $\tau \in \omega^{<\omega}$, if $\sigma \in T$ and $\tau \preceq \sigma$, then $\tau \in T$. A *path* in T is a function $f \in \omega^\omega$ such that the set

$$\text{(5.34)} \qquad f[n] = \{\, \sigma \in \omega^{<\omega} \mid f(i) = \sigma(i), i < |\sigma| \,\}$$

is a subset of T.

For each $e \in \omega$, let T_e be the tree defined by

$$\text{(5.35)} \qquad \sigma \in T_e \quad \Leftrightarrow \quad \{e\}(\sigma(i), \sigma(i-1))\downarrow \text{ for all } i < |\sigma|$$

It is clear from the discussion in Sacks [43] on the analysis of Π^1_1 sets that the predicate

$$\text{(5.36)} \qquad \mathcal{R}(e) =_{\mathrm{def}} T_e \text{ has no path}$$

is Π^1_1-complete. So we need only construct a uniformly computable sequence of structures $\mathfrak{A}_e$ over $\mathcal{L}$ such that

$$(5.37)\qquad T_e \text{ has no path} \quad\Leftrightarrow\quad \mathfrak{A}_e \text{ is saturated}$$

By appropriately coding $\omega^{<\omega}$ in ω, we can label our unary predicates P_n from now on as P_σ for $\sigma \in \omega^{<\omega}$. $\mathfrak{A}_e$ will have domain $\omega^{<\omega} \times \omega$. For each e, $s \in \omega$, we let T_e^s be the approximation of T_e at stage s. In other words,

$$(5.38)\qquad \sigma \in T_e^s \quad\Leftrightarrow\quad \{e\}_s(\sigma(i), \sigma(i-1))\downarrow \text{ for all } i < |\sigma|$$

Then the sets T_e^s are uniformly computable. So, we can define a uniformly computable sequence of structures $\mathfrak{A}_e$ by defining, for each $\langle \sigma, s\rangle \in \mathfrak{M}_e$ and $\tau \in \omega^{<\omega}$,

$$(5.39)\qquad \mathfrak{A}_e \models P_\tau(\langle \sigma, s\rangle) \quad\Leftrightarrow\quad \tau \preceq \sigma \text{ and } \sigma \in T_e^s$$

We need only show that these structures have property (5.37).

Take any complete 1-type $\Gamma(x, \overline{a})$ over $\mathfrak{A}_e$. If $(x = a_i) \in \Gamma(x, \overline{a})$ for some i, then this type is trivially realized in $\mathfrak{M}_e$. Otherwise, since all of the predicates in our language are unary, Γ is uniquely determined by the set

$$(5.40)\qquad \Phi(\overline{x}) = \{\, P_\sigma(x) \in \Gamma(x, \overline{a}) \,\} \cup \{\, \neg P_\tau(x) \in \Gamma(x, \overline{a}) \,\} \cup \{\, (x \neq a_i) \,\}$$

So this means that we need only consider sets of formulae of the form

$$(5.41)\qquad \Gamma_{X,\overline{a}}(x) = \{\, P_\sigma(x) \mid \sigma \in X \,\} \cup \{\, \neg P_\tau(x) \mid \tau \notin X \,\} \cup \{\, (x \neq a_i) \,\}$$

for $X \subseteq \omega^{<\omega}$.

Definition 5.3.3. For any $X \subseteq \omega^{<\omega}$, we say that X is a *finite branch of T_e* if $X = \emptyset$ or there is some $\sigma \in T_e$ such that for all $\tau \in \omega^{<\omega}$

$$(5.42)\qquad \tau \in X \quad\Leftrightarrow\quad \tau \preceq \sigma$$

Lemma 5.3.4. *For all $X \subseteq \omega^{<\omega}$, and $\overline{a} \in \mathfrak{A}_e$, $\Gamma_{X,\overline{a}}$ is realized in $\mathfrak{A}_e$ if and only if X is a finite branch of T_e.*

Proof. ($\Rightarrow$) Suppose $\Gamma_{X,\overline{a}}$ is realized by $\langle \sigma, s\rangle \in \mathfrak{A}_e$. If $\sigma \notin T_e^s$, then $X = \emptyset$ by (5.39). Otherwise, (5.42) follows from (5.39).

($\Leftarrow$) Let X be any finite branch of T_e with $\overline{a} \in \mathfrak{A}_e$. If $X = \emptyset$, then by (5.39), any element $\langle \sigma, 0\rangle \notin \overline{a}$ realizes $\Gamma_{X,\overline{a}}$, since $T_e^0 = \emptyset$.

If $X \neq \emptyset$, let $\sigma \in T_e$ be the element defining X in (5.42). There is some $t \in \omega$ such that $\sigma \in T_e^s$ for all stages $s \geq t$. Let $s > t$ be such that $s > r$ for all $\langle \rho, r\rangle \in \overline{a}$. Then $\langle \sigma, s\rangle \notin \overline{a}$ and for all $\tau \in \omega^{<\omega}$

$$(5.43)\qquad \mathfrak{A}_e \models P_\tau(\langle \sigma, s\rangle) \quad\Leftrightarrow\quad \tau \preceq \sigma \quad\Leftrightarrow\quad \tau \in X$$

So $\langle \sigma, s\rangle \notin \overline{a}$ realizes $\Gamma_{X,\overline{a}}$ by (5.39). ■

Lemma 5.3.5. *For all $e \in \omega$, T_e has no path if and only if for every $\Gamma_{X,\overline{a}}$ finitely realizable $\mathfrak{A}_e$, X is a finite branch of T_e.*

Proof. ($\Rightarrow$) Let $\Gamma_{X,\overline{a}}$ be consistent with $\mathfrak{A}_e$. First, suppose for a contradiction that there is no n such that $\tau \notin X$ for all $|\tau| > n$. Take any $\sigma_0 \in X$. Let $\sigma_1 \in X$ be any element with $|\sigma_1| > |\sigma_0|$. As $\Gamma_{X,\overline{a}}$ is consistent with $\mathfrak{A}_e$, there is some $\langle \tau, s\rangle \in \mathfrak{A}_e$ such that

$$\mathfrak{A}_e \models P_{\sigma_0}(\langle \tau, s\rangle) \wedge P_{\sigma_1}(\langle \tau, s\rangle) \tag{5.44}$$

Then (5.39) implies that $\sigma_0, \sigma_1 \preceq \rho$. Therefore, since $|\sigma_1| > |\sigma_0|$, we have that $\sigma_0 \prec \sigma_1$.

Furthermore, since $\rho \in T_e^s$, we have that σ_0, $\sigma_1 \in T_e$. Continuing by induction we can define an infinite sequence of elements $\sigma_n \in T_e$ such that $\sigma_n \prec \sigma_{n+1}$ for all n. This determines a path in T_e, which is a contradiction.

Now suppose there is some n such that $\tau \notin X$ for all $|\tau| > n$. As $\Gamma_{X,\overline{a}}$ is finitely realizable $\mathfrak{A}_e$, let $\langle \sigma, s\rangle \in \mathfrak{A}_e$ be such that

$$\mathfrak{A}_e \models P_\tau(\langle \sigma, s\rangle) \quad \Leftrightarrow \quad \tau \in X \tag{5.45}$$

for all $|\tau| \leq n$. So, by (5.39), we have that

$$\tau \in X \quad \Leftrightarrow \quad \tau \preceq \sigma \tag{5.46}$$

for all $|\tau| \leq n$. As X does not contain any elements $|\tau| > n$ and $\sigma \in T_e^s$, X is clearly a finite branch of T_e.

($\Leftarrow$) Suppose that T_e has an path f. Let $X = f[n]$ and consider the set of formulae $\Gamma_X(x)$. Take any $\sigma \in X$. There is some $s \in \omega$ such that $\sigma \in T_e^s$. So we have that (5.43) holds for all $|\tau| \leq |\sigma|$. As this is true for arbitrary $\sigma \in X$, $\Gamma_X(x)$ is finitely realizable in $\mathfrak{A}_e$. As X is infinite, it cannot be a finite branch. ■

Our result is immediate from the lemmas above. □

Unfortunately, as Theorem 5.2.5 illustrates, this result does not necessarily mean that saturation is Π^1_1-complete in the language with a single binary relation. Fortunately, the following result in Chapter 5 of Peretyat'kin [38], allows us to translate Theorem 5.3.1 to the this language.

Theorem 5.3.6 (Peretyat'kin [38]). *Let $\mathcal{L}_0$ be a computable language with infinitely many relations, $\mathcal{L}_1$ the language of a single binary relation R. For any countable structure $\mathfrak{A}$ of $\mathcal{L}_0$, there is a structure* $\mathrm{IG}(\mathfrak{A})$*, uniformly computable in* $\mathrm{diag}(\mathfrak{A})$*, such that $\mathfrak{A}$ is saturated if and only if* $\mathrm{IG}(\mathfrak{A})$ *is.*

Corollary 5.3.7. *Saturation is Π^1_1-complete.*

As we mentioned earlier, this result implies that there is no computable sentence $\varphi_{\mathrm{SAT}} \in \mathcal{L}_{\omega_1,\omega}$ such that a structure $\mathfrak{A}$ is saturated if and only if $\mathfrak{A} \models \varphi_{\mathrm{SAT}}$. This result still leaves open the possibility that there is a noncomputable element of $\mathcal{L}_{\omega_1,\omega}$ that defines saturation. However, as we shall see in section 5.6, this is not the case.

5.4 Existence of isomorphisms

We know from a result of Scott [44] that, for any structure $\mathfrak{A}$, there is a sentence $\varphi_{\mathfrak{A}} \in \mathcal{L}_{\omega_1,\omega}$ such that

$$\mathfrak{B} \models \varphi_{\mathfrak{A}} \quad \Leftrightarrow \quad \mathfrak{A} \cong \mathfrak{B} \tag{5.47}$$

We know from Ash and Knight [2] that, for a computable structure $\mathfrak{A}$, $\varphi_{\mathfrak{A}}$ is a Σ_α sentence, with $\alpha \leq \omega_1^{\mathrm{CK}}$. Furthermore, $\varphi_{\mathfrak{A}}$ is a computable Σ_α sentence when $\alpha < \omega_1^{\mathrm{CK}}$.

An interesting question is whether or not there is a computable structure $\mathfrak{A}$ for which $\varphi_{\mathfrak{A}}$ is not a computable infinitary sentence. For each structure $\mathfrak{A}$, we define the predicate

$$\mathrm{ISOM}_{\mathfrak{A}}(e) =_{\mathrm{def}} (\mathfrak{M}_e \cong \mathfrak{A}) \tag{5.48}$$

The complexity of this predicate depends on the choice of $\mathfrak{A}$. However, for computable structures, as with AUT in (5.1), it is clear that this predicate is Σ^1_1. Furthermore, if $\varphi_{\mathfrak{A}}$ is a computable infinitary sentence, then this predicate is hyperarithmetic. So, for any computable structure $\mathfrak{A}$ such that $\mathrm{ISOM}_{\mathfrak{A}}$ is Σ^1_1-complete, $\varphi_{\mathfrak{A}}$ can not be computable. Our goal of this section is to construct such a computable structure.

To do this, we want to construct from each Σ^1_1 predicate $\mathcal{P}(n)$ a uniformly computable sequence of structures $\mathfrak{A}_n$ which are all isomorphic for the n such that $\mathcal{P}(n)$ holds. This is exactly what we do in the following proposition.

Proposition 5.4.1. *Let $\mathcal{P}(n)$ be a Π^1_1 predicate. There is a uniformly computable sequence of linear orderings $\mathfrak{A}_n$ such that for all $n \in \omega$,*

$$\mathfrak{A}_n \cong \begin{cases} \alpha, \textit{ for some } \alpha < \omega_1^{\mathrm{CK}} & \textit{if } P(n) \\ \omega_1^{\mathrm{CK}}(1+\eta) & \textit{otherwise} \end{cases} \tag{5.49}$$

where η is the order type of $(\mathbb{Q}, \leq)$.

Proof. Let $\mathcal{P}(n) = \forall f\, \mathcal{Q}(f,n)$, where $\mathcal{Q}$ is arithmetic. We define the predicate

$$\mathcal{S}(f,n) =_{\mathrm{def}} \big(\neg\mathcal{Q}(f_0,n) \wedge (f_1 \notin \mathrm{HYP})\big) \tag{5.50}$$

where $f_i(x) = y$ for $f(x) = \langle i, y\rangle$. We know from Kleene [32] that the predicate $(f \in \mathrm{HYP})$ is Π^1_1. Hence $\mathcal{S}$ is a Σ^1_1 predicate. Furthermore, it is clear that $\mathcal{P}(n)$ holds if and only if $\forall f\, \neg\mathcal{S}(f,n)$.

For any function $g : \omega \to \omega$ and $n \in \omega$, we let

$$\overline{g}(n) = \langle\langle 0, g(0)\rangle, \ldots, \langle n-1, g(n-1)\rangle\rangle \tag{5.51}$$

Using the normal forms from Kleene [32], we can write $\mathcal{S}$ as $\exists g\, \forall x\, \mathcal{R}(\overline{f}(x), \overline{g}(x), n)$, with $\mathcal{R}$ computable. We let $\mathfrak{B}_n$ be the Kleene-Brouwer ordering of $\mathcal{R}$. That is,

$$\mathrm{dom}(\mathfrak{B}_n) = \{\, \sigma \mid \forall \tau \preceq \sigma\, \mathcal{R}(\tau_0, \tau_1, n), \tau_i(k) = m \text{ when } \tau(k) = \langle i, m\rangle \,\} \tag{5.52}$$

ordered by the relation $\sigma \leq_{\mathcal{R}} \tau$ where either

(a) $|\sigma| \geq |\tau|$ and $\tau \preceq \sigma$, or

(b) There is some $k < |\sigma|, |\tau|$ such that $\sigma(k) < \tau(k)$ and σ, τ agree up to k.

This is a computable linear ordering since $\mathcal{R}$ is computable.

Finally, we let $\mathfrak{A}_n = \oplus_\omega \mathfrak{B}_n$ be the direct sum of countably many copies of $\mathfrak{B}_n$. In other words, $\mathfrak{A}_n = \bigcup_{k\in\omega} \mathfrak{A}_n^k$ where $\mathfrak{A}_n^k \cong \mathfrak{B}_n$ for all $k \in \omega$ and

$$\mathfrak{A}_n \vDash (a \leq b) \quad \Leftrightarrow \quad \begin{cases} \mathfrak{A}_n^k \vDash (a \leq b) & \text{if } a,\, b \in \mathfrak{A}_n^k \\ k < \ell & \text{if } a \in \mathfrak{A}_n^k,\, b \in \mathfrak{A}_n^\ell, \text{ for } k \neq \ell \end{cases} \tag{5.53}$$

This is also a computable linear ordering since $\mathfrak{B}_n$ is.

Suppose now that $\{\sigma_k\}$ is an infinite descending sequence of $\mathfrak{B}_n$. For any $\sigma \in \mathfrak{B}_n$, there are only finitely many elements $\tau \leq_{\mathcal{R}} \sigma$ by condition (b). So we can assume without loss of generality that the σ_k are ordered by condition (a). By the definition of the domain of $\mathfrak{B}_n$ in (5.52), we can add elements to this sequence so that there is some function $h : \omega \to \omega$ with $h[k] = \{\sigma_n\}$. Then $\forall x\, \mathcal{R}(\overline{h_0}(x), \overline{h_1}(x), n)$, and hence $\mathcal{S}(h_0, n)$ holds. The function h cannot be hyperarithmetic, since $(h_0)_1 \notin \mathrm{HYP}$. So by a result of Harrison [21], $\mathfrak{B}_n$ has order type $\omega_1^{\mathrm{CK}}(1+\eta)+\gamma_n$, for some computable ordinal γ_n. It is clear then from the definition of $\mathfrak{A}_n$ that $\mathfrak{A}_n \cong \omega_1^{\mathrm{CK}}(1+\eta)$.

If $\mathfrak{B}_n$ is a well-ordering, then so is $\mathfrak{A}_n$. Hence we have just shown that

$$\mathfrak{A}_n \cong \begin{cases} \alpha, \text{ for some } \alpha < \omega_1^{\mathrm{CK}} & \text{if } \mathfrak{B}_n \text{ is well-ordered} \\ \omega_1^{\mathrm{CK}}(1+\eta) & \text{otherwise} \end{cases} \tag{5.54}$$

So we need only show now that $\mathfrak{B}_n$ is a well-ordering if and only if $\forall f\, \neg\mathcal{S}(f, n)$. If $h[n]$ is an infinite descending sequence in $\mathfrak{B}_n$, we have already seen that $\mathcal{S}(h_0, n)$ holds. Otherwise, let f be any function such that $\mathcal{S}(f, n)$ holds. Then it follows from condition (a) that there is some function h such that $h_0 = f$ and $h[n]$ is an infinite descending sequence in $\mathfrak{B}_n$. □

This proposition gives us the reduction necessary to prove the following result.

Theorem 5.4.2. *There is a computable structure $\mathfrak{B}$ such that $\mathrm{ISOM}_{\mathfrak{B}}$ is Σ_1^1-complete.*

Proof. We let $\mathfrak{B}$ be a computable presentation of $\omega_1^{\mathrm{CK}}(1 + \eta)$. There is such a presentation since we know from Harrison [21] that there is some computable ordinal α such that $\omega_1^{\mathrm{CK}}(1+\eta) + \alpha$ is computably presentable, and the direct sum of countably many copies of this presentation produces a computable presentation of $\omega_1^{\mathrm{CK}}(1+\eta)$.

Let $\mathcal{P}(n)$ be a Π_1^1 predicate. As before, we need to construct a uniformly computable sequence of structures $\mathfrak{A}_n$ such that $\mathfrak{A}_n \cong \mathfrak{B}$ if and only if $\mathcal{P}(n)$ holds. It is clear that the sequence from Proposition 5.4.1 is sufficient. □

While we have been focusing our attention on properties of specific computable structures, we can also analyze the complexity of relations between computable structures. For example, we could look at the complexity of the predicate

$$\mathrm{ISOM}(e,i) =_{\mathrm{def}} (\mathfrak{M}_e \cong \mathfrak{M}_i) \tag{5.55}$$

The following is immediate from Theorem 5.4.2.

Corollary 5.4.3. [1] ISOM *is* Σ^1_1*-complete.*

Since isomorphism is a relationship between structures, and not a property of a specific structure, it does not make sense to talk of a computable infinitary sentence which "defines" isomorphism. Therefore, the completeness result in Corollary 5.4.3 does not really an anticharacterization in the same way that Theorem 5.4.2 is. However, Corollary 5.4.3 is important in that it can help us analyze the complexity of properties that are defined in the context of isomorphisms. The properties of hyperarithmetic stability and hyperarithmetic categoricity in the next section are two such properties.

Another property that is defined using isomorphisms is rigidity.

Definition 5.4.4. A structure $\mathfrak{A}$ is *rigid* if the only automorphism of $\mathfrak{A}$ is the identity.

It is clear that any ordinal is rigid. For any ordinal α, we can prove by induction on $\beta < \alpha$ that any automorphism of α fixes β. This fact gives us the following result.

Theorem 5.4.5. *Rigidity is* Π^1_1*-complete.*

Proof. Rigidity is at most Π^1_1, as it is defined by the predicate

$$\mathrm{RIG}(e) =_{\mathrm{def}} \forall f\, \big((f : \mathfrak{M}_e \cong \mathfrak{M}_e) \to \forall x(f(x) = x)\big) \tag{5.56}$$

Let $\mathfrak{A}_n$ be the uniformly computable sequence of structures from Proposition 5.4.1. As any computable ordinal is rigid, we need only show that $\omega_1^{\mathrm{CK}}(1+\eta)$ is not rigid. Clearly η is not rigid, so let g be a nontrivial automorphism η. This induces a nontrivial automorphism f of $\omega_1^{\mathrm{CK}}(1+\eta)$ in the obvious manner. □

5.5 Categoricity and stability

As we saw in Theorem 4.7.4, we are often interested in the complexity of the maps between computable structures. This fact motivates the following definition.

[1] This result was independently discovered by Friedman and Stanley [13], using essentially the same proof.

Definition 5.5.1. Let Γ be a collection of subsets of ω. A computable structure $\mathfrak{A}$ is Γ-*categorical* if for any computable structure $\mathfrak{B} \cong \mathfrak{A}$, there is an isomorphism $f : \mathfrak{A} \to \mathfrak{B}$ with $f \in \Gamma$. A computable structure $\mathfrak{A}$ is Γ-*stable* if for any computable structure $\mathfrak{B}$ and any isomorphism $f : \mathfrak{A} \to \mathfrak{B}$, $f \in \Gamma$.

The case when Γ is the collection of hyperarithmetic sets is of particular interest to computable model theory, and has been well-studied by Ash [1]. Hyperarithmetic stability is equivalent to the following predicate

$$\text{(5.57)} \qquad \mathrm{STAB}_{\mathrm{HYP}}(e) =_{\mathrm{def}} \forall f \ \forall n \left((f : \mathfrak{M}_e \cong \mathfrak{M}_n) \to \exists e \, \exists b \in O \left(\{e\}^{H_b} = f\right)\right)$$

where O is Kleene's set of notations for the computable ordinals and H_b are the H-sets defined in Sacks [43]. The predicate $(x \in O)$ is Π^1_1-complete by a result of Kleene (Reference). Therefore, this predicate is at most Π^1_1. Similarly, hyperarithmetic categoricity is equivalent to the Π^1_1 predicate

$$\text{(5.58)} \quad \mathrm{CAT}_{\mathrm{HYP}}(e) =_{\mathrm{def}} \forall n \big(\exists f (f : \mathfrak{M}_e \cong \mathfrak{M}_n) \to \exists e \exists b \in O(\{e\}^{H_b} : \mathfrak{M}_e \cong \mathfrak{M}_n)\big)$$

Our goal of this section is to show that both of these properties are Π^1_1-complete.

So that we can handle both of these properties simultaneously, we introduce the following definition.

Definition 5.5.2. For any predicates $\mathcal{P}_i$, $\mathcal{Q}_i$ on ω with $i \in \{0,1\}$, we say that $(\mathcal{P}_0, \mathcal{P}_1) \leq_{1:1} (\mathcal{Q}_0, \mathcal{Q}_1)$ if there is some computable, one-one function $f : \omega \to \omega$ such that, for all $e \in \omega$,

(a) $\mathcal{P}_i(e)$ implies $\mathcal{Q}_i(f(e))$ for $i \in \{0,1\}$.

(b) $\neg\mathcal{P}_0(e) \wedge \neg\mathcal{P}_1(e)$ implies $\neg\mathcal{Q}_0(f(e)) \wedge \neg\mathcal{Q}_1(f(e))$.

When Γ_i are collections of subsets of ω, we define $(\Gamma_0, \Gamma_1) \leq_{1:1} (\mathcal{Q}_0, \mathcal{Q}_1)$ as in Definition 5.2.2.

Suppose that $\mathcal{Q}_0$ and $\mathcal{Q}_1$ are disjoint relations. It is clear in this case that $(\Gamma_0, \Gamma_1) \leq_{1:1} (\mathcal{Q}_0, \mathcal{Q}_1)$ implies $\Gamma_i \leq_{1:1} \mathcal{Q}_i$ for $i \in \{0,1\}$. So we need only prove the following result.

Theorem 5.5.3. $(\Pi^1_1, \Sigma^1_1) \leq_{1:1} (\mathrm{STAB}_{\mathrm{HYP}}, \neg\mathrm{CAT}_{\mathrm{HYP}})$.

Proof. Let $\mathcal{P}(n)$ be a Π^1_1 predicate. We need to construct a uniformly computable sequence of structures $\mathfrak{A}_n$ such that $\mathfrak{A}_n$ is hyperarithmetically stable if $\mathcal{P}(n)$ holds, and not hyperarithmetically categorical otherwise.

We let the structures $\mathfrak{A}_n$ be the uniformly computable sequence from Proposition 5.4.1. Ash [1] has shown that the computable ordinals are all hyperarithmetically stable. So we need only show that, whenever $\mathfrak{A}_n \cong \omega_1^{\mathrm{CK}}(1+\eta)$, $\mathfrak{A}_n$ is not Δ^0_α-categorical for any computable ordinal α.

Let α be a computable ordinal, and suppose for a contradiction that $\mathfrak{A} \cong \omega_1^{\mathrm{CK}}(1+\eta)$ is Δ^0_α-categorical. Again from Ash [1] we know that there is some computable ordinal δ that is not Δ^0_α-stable. Any ordinal is rigid; that is, it has no nontrivial automorphisms. So δ cannot be Δ^0_α-categorical either.

Let $\mathfrak{B}$ be the standard computable presentation of δ. Let $\mathfrak{C} = \mathfrak{B} \oplus \mathfrak{A}_n$. That is, $\mathfrak{C}$ is the computable direct sum of $\mathfrak{B}$ and $\mathfrak{A}_n$, where all of the elements of $\mathfrak{A}_n$ are greater than the elements of $\mathfrak{B}$. Then

$$\mathfrak{C} \cong \delta \oplus \omega_1^{\mathrm{CK}}(1+\eta) \cong \omega_1^{\mathrm{CK}}(1+\eta) \cong \mathfrak{A}_n \tag{5.59}$$

Now let $\mathfrak{B}'$ be any other computable presentation of δ and define $\mathfrak{C}'$ similarly. The Δ^0_α bijections are closed under inverse and composition. Therefore, since $\mathfrak{A}_n$ is Δ^0_α-categorical, any computable presentation of $\mathfrak{A}_n$ is Δ^0_α-categorical. Hence there is a Δ^0_α isomorphism $f : \mathfrak{C} \to \mathfrak{C}'$. Let $f_{\mathfrak{B}}$ be the restriction of f to $\mathfrak{B} \subseteq \mathfrak{C}$. Since the domain of $\mathfrak{B}$ is a computable subset of $\mathfrak{C}$, $f_{\mathfrak{B}}$ is also Δ^0_α.

Since f is a bijection preserving order, the image of an initial segment of $\mathfrak{C}$ must be an initial segment of $\mathfrak{C}'$. Therefore, by the rigidity of ordinals, $f_{\mathfrak{B}}(\mathfrak{B}) = \mathfrak{B}'$ and hence $f_{\mathfrak{B}} : \mathfrak{B} \to \mathfrak{B}'$ is an isomorphism. As $\mathfrak{B}'$ was an arbitrary computable presentation of δ, this means that δ is Δ^0_α-categorical, which is a contradiction. ☐

The rigidity of ordinals is important in this final argument. In order to know that $f_{\mathfrak{B}}(\mathfrak{B}) = \mathfrak{B}'$, we needed that for any ordinal δ, there is a unique initial segment of $\omega_1^{\mathrm{CK}}(1+\eta)$ with order type δ. This is not true for general initial segments; there are many initial segments of $\omega_1^{\mathrm{CK}}(1+\eta)$ with order type $\omega_1^{\mathrm{CK}}(1+\eta)$.

5.6 Applications to descriptive set theory

A possible complaint about the analysis in this chapter is that we have restricted ourselves computable structures. Only the properties of hyperarithmetic categoricity and stability are unique to computable structures; the others are defined for all countable structures. As computable structures are the major focus of this work, this is not that important an issue. Furthermore, so long as we view these completeness results as anticharacterizations, they hold for all structures. If two definitions are not equivalent over computable structures, they are not equivalent over any larger class of structures.

However, for aesthetic reasons, we may want to analyze the complexity of these properties over all countable structures. We know what it means for a predicate $\mathcal{P}(f)$ on ω^ω to be Σ^1_1 or Π^1_1. We also showed above in (5.3) that there is a correspondence between elements of 2^ω and countable structures. Moreover, we know from (5.4) that this correspondence gives all countable structures for our language. So we could define the complexity of a property with respect to countable structures in the usual way.

If we do this, it is clear that for all of the properties investigated in this chapter, the upper bounds on the complexity remain the same. Unfortunately, our technique for computing lower bounds only applies to predicates on ω. If we want a similar technique for computing lower bounds for predicates on ω^ω, we will need a new collection of complexity classes.

We turn to the field of descriptive set theory for our new complexity classes. In descriptive set theory, we take each of the complexity classes of the analytic and hyperarithmetic hierarchies and define boldface versions of them.

Definition 5.6.1. Let Γ be a collection of predicates on ω^ω. The *boldface version of* Γ, which we denote $\utilde{\Gamma}$, is the collection of all predicates $\utilde{\mathcal{P}}(f)$ such that $\utilde{\mathcal{P}}$ is Γ in some $g \in \omega^\omega$(i.e. $\utilde{\mathcal{P}}(f) = \mathcal{P}(f, g)$ for $\mathcal{P} \in \Gamma$). We call this element $g \in \omega^\omega$ a *parameter* of $\utilde{\mathcal{P}}$.

We say that a predicate $\utilde{\mathcal{P}}(f)$ on ω^ω is $\utilde{\Gamma}$-*hard* if for all $\utilde{\mathcal{Q}}(f) \in \utilde{\Gamma}$ with parameter $g \in \omega^\omega$, there is a $e \in \omega$ such that for all $f \in \omega^\omega$, $\utilde{\mathcal{Q}}(f)$ if and only if $\utilde{\mathcal{P}}(\{e\}^{f \oplus g})$. We say that $\utilde{\mathcal{P}}(f)$ is $\utilde{\Gamma}$-*complete* if, in addition, $\utilde{\mathcal{P}} \in \utilde{\Gamma}$.

The work by Kechris [29] is an excellent survey of the field of descriptive set theory. Similar arguments to the ones for predicates on ω show that these new reductions determine lower bounds in the boldface complexity classes. So we define the complexity of a model theoretic property $\mathcal{P}$ over countable structures to be the appropriate boldface complexity class for the predicate

$$\mathbf{MOD}_{\mathcal{P}}(f) =_{\text{def}} \mathfrak{M}_f \text{ has property } \mathcal{P} \tag{5.60}$$

In the completeness results for computable structures above, our reductions consisted of a uniformly computable sequence of structures $\mathfrak{A}_n$ such that, for our property $\mathcal{P}$ and predicate $\mathcal{R}(n)$

$$\mathfrak{A}_n \text{ has property } \mathcal{P} \quad \Leftrightarrow \quad \mathcal{R}(n) \tag{5.61}$$

By Definition 5.6.1, we can prove completeness results for model theoretic properties in descriptive set theory in much the same way. Given some boldface predicate $\utilde{\mathcal{R}}(f)$ with parameter $g \in \omega^\omega$, we want to construct structures $\mathfrak{A}_f$, uniformly computable in f and g, such that

$$\mathfrak{A}_f \text{ has property } \mathcal{P} \quad \Leftrightarrow \quad \utilde{\mathcal{R}}(f) \tag{5.62}$$

It is clear that, if we can construct such structures for any $\utilde{\mathcal{R}} \in \utilde{\Gamma}$, then property $\mathcal{P}$ is $\utilde{\Gamma}$-hard.

Given this observation, we can easily adapt the proof of Corollary 5.3.7 to descriptive set theory.

Theorem 5.6.2 (Boldface Corollary 5.3.7). *Saturation is* $\utilde{\Pi}^1_1$*-complete.*

Proof. For each $f \in 2^\omega$, let T_f be the tree defined by

$$\sigma \in T_f \quad \Leftrightarrow \quad f(\sigma(i), \sigma(i-1)) = 1 \text{ for all } i < |\sigma| \tag{5.63}$$

This tree is computable in f. Again, we know from the discussion in Sacks [43] on the analysis of Π^1_1 sets that the Π^1_1 predicate

$$\mathbf{PATH}(f) =_{\text{def}} T_f \text{ has no path} \tag{5.64}$$

is $\underset{\sim}{\Pi}^1_1$-complete. So we need only construct structures $\mathfrak{A}_f$, uniformly computable in f such that $\mathfrak{A}_f$ is saturated exactly when T_f has no path.

We let $\mathcal{L}$ be the language of infinitely many unary predicates P_σ, $\sigma \in \omega^{<\omega}$ from Theorem 5.3.1. For each $f \in \omega^\omega$, we define the structure

$$\mathfrak{A}_f \models P_\tau(\langle\sigma\rangle) \quad \Leftrightarrow \quad \tau \preceq \sigma \text{ and } \sigma \in T_f \tag{5.65}$$

Since the T_f are uniformly computable in f, so are the structures $\mathfrak{A}_f$. So it is easy to see that the structures $\mathfrak{A}_f$ witness that saturation is $\underset{\sim}{\Pi}^1_1$-complete in this language. We extend this result to the language of a single binary relation by Theorem 5.3.6. □

As we saw above, Corollary 5.3.7 allowed us to conclude that there is no computable infinitary formula defining saturation. Because of the following result, Theorem 5.6.2 allows us to conclude that there is no element of $\mathcal{L}_{\omega_1,\omega}$ defining saturation.

Definition 5.6.3. Let θ_n be an enumeration of the atomic formulae of $\mathcal{L}$. For each $f \in \omega^\omega$, we can associate a formula $\phi_f \in \mathcal{L}_{\omega_1,\omega}$ as follows.

(a) If $f(0) = 0$, then $\phi_f = \theta_{f(1)}$.

(b) If $f(0) = 1$, then $\phi_f = \neg\phi_g$, where $g(n) = f(n+1)$.

(c) If $f(0) = 2$, then $\phi_f = \bigwedge_{g_k} \phi_{g_k}$ where $g_k(n) = f(\langle k, n\rangle + 1)$.

(d) Suppose $f(0) = 3 + \lfloor\overline{x}\rfloor$, where $\lfloor\overline{x}\rfloor \in \omega$ uniquely codes the sequence of variables $\overline{x}$. Then $\phi_f = \exists\overline{x}\phi_g(\overline{x})$, where $g(n) = f(n+1)$.

It is clear from this definition that any formula $\varphi \in \mathcal{L}_{\omega_1,\omega}$ is equivalent to some ϕ_f, $f \in \omega^\omega$.

Proposition 5.6.4. *For any $g \in \omega^\omega$, the predicate*

$$\mathrm{MOD}_{\phi_g}(f, \overline{a}) =_{\text{def}} \mathfrak{M}_f \models \phi_g(\overline{a}) \tag{5.66}$$

is uniformly Δ^1_1 in g.

Proof. We proceed by induction on the complexity of ϕ_g.

Case $\phi_g \in \mathcal{L}^{\text{atm}}$: In this case, MOD_{ϕ_g} is uniformly computable from the value $g(1)$ by the definition of $\mathfrak{M}_f$ in (5.3).

Case $\phi_g = \neg\phi_h$: Then $\mathrm{MOD}_{\phi_g} = \neg\mathrm{MOD}_{\phi_h}$. Then MOD_{ϕ_g} is uniformly Δ^1_1 in g, since MOD_{ϕ_h} is uniformly Δ^1_1 in h.

Case $\phi_g = \bigwedge_{h_n} \phi_{h_n}$: Then

$$\mathrm{MOD}_{\phi_g}(f, \overline{a}) = \forall n \, \mathrm{MOD}_{\phi_{h_n}}(f, \overline{a}) \tag{5.67}$$

We know that Δ^1_1 is closed under quantifiers over ω. Hence, as the $\mathrm{MOD}_{\phi_{h_n}}$ are all uniformly Δ^1_1 in h_n, we have that MOD_{ϕ_g} is uniformly Δ^1_1 in g.

Case $\phi_g = \exists \overline{x} \, \phi_h(\overline{x})$: Then

$$\mathrm{MOD}_{\phi_g}(f, \overline{a}) = \exists \overline{b} \, \mathrm{MOD}_{\phi_h}(f, \overline{b}\overline{a}) \tag{5.68}$$

Again, we know that Δ^1_1 is closed under quantifiers over $\omega^{<\omega}$. So MOD_{ϕ_g} is uniformly Δ^1_1 in g, since MOD_{ϕ_h} is uniformly Δ^1_1 in h. □

Suppose now that saturation is defined by an infinitary sentence $\varphi \in \mathcal{L}_{\omega_1,\omega}$, and let $g \in \omega^\omega$ be such that φ and ϕ_g are logically equivalent. Then by Proposition 5.6.4, saturation must be Δ^1_1 in g, which contradicts Theorem 5.6.2.

To adapt the results in section 5.4 to descriptive set theory, we need only modify Proposition 5.4.1.

Proposition 5.6.5 (Boldface Proposition 5.4.1). *Let $\mathcal{P}(f,g)$ be Π^1_1 in $g \in \omega^\omega$. For each $f \in \omega^\omega$, there is a linear ordering $\mathfrak{A}_f$, uniformly computable in f and g, such that*

$$\mathfrak{A}_f \cong \begin{cases} \alpha, \text{ for some } \alpha < \omega_1^{f\oplus g} & \text{if } \mathcal{P}(f,g) \\ \omega_1^{f\oplus g}(1+\eta) & \text{otherwise} \end{cases} \tag{5.69}$$

where η is the order type of $(\mathbb{Q}, \leq)$.

Proof. Let $\mathcal{P}(f) = \forall h \, \mathcal{Q}(f,g,h)$, where $\mathcal{Q}$ is arithmetic. Again we define the predicate

$$\mathcal{S}(f,g,h) =_{\mathrm{def}} \big(\neg\mathcal{Q}(f,g,h_0) \wedge (h_1 \notin \mathrm{HYP}^{f\oplus g})\big) \tag{5.70}$$

The result of Kleene [32] relativizes to $\mathrm{HYP}^{f\oplus g}$, so again $\mathcal{S}$ is Π^1_1.

Using the normal forms of Kleene [32], we contruct some computable relation $\mathcal{R}$ such that

$$\mathcal{S}(f,g,h) \quad \Leftrightarrow \quad \exists k \, \forall x \, \mathcal{R}(\overline{k}(x), \overline{f}(x), \overline{g}(x), \overline{h}(x)) \tag{5.71}$$

We let $\mathfrak{B}_f$ be the Kleene-Brouwer ordering on $\mathcal{R}$, and define $\mathfrak{A}_f = \oplus_\omega \mathfrak{B}_f$. These structures are uniformly computable in f and g since the Kleene-Brouwer ordering of $\mathcal{R}$ is. Furthermore, since we can relativize the result of Harrison [21] to ordinals computable in $f \oplus g$, these structures satisfy (5.69). □

We can immediately use this result to prove a descriptive set theory analogue of Theorem 5.4.5.

Theorem 5.6.6. *Rigidity is $\utilde{\Pi}^1_1$-complete.*

Proof. All ordinals are rigid. For any $f,\ g \in \omega^\omega$, $\omega_1^{f \oplus g}(1+\eta)$ is not rigid for the same reasons as in Theorem 5.4.5. $\square$

However, we cannot use Proposition 5.6.5 to prove a descriptive set theory analogue of Theorem 5.4.2. In Proposition 5.4.1, the Σ^1_1 predicate $\mathcal{P}$ produces the structure $\omega_1^{\mathrm{CK}}(1+\eta)$ whenever $\mathcal{P}(n)$ holds. However, for any $\utilde{\Sigma}^1_1$ predicate $\utilde{\mathcal{P}}(f)$ with parameter $g \in \omega$, the linear order $\omega_1^{f \oplus g}(1+\eta)$ depends on both f and g. So the proof in Theorem 5.4.2 cannot be adapted to descriptive set theory.

Furthermore, we know that for any structure $\mathfrak{A}$, there is some sentence $\varphi_{\mathfrak{A}} \in \mathcal{L}_{\omega_1,\omega}$ as in (5.47). So by Proposition 5.6.4, the predicate

$$\mathbf{ISO}_{\mathfrak{A}}(f) =_{\text{def}} (\mathfrak{M}_f \cong \mathfrak{A}) \tag{5.72}$$

is $\utilde{\Delta}^1_1$. However, if we look at isomorphism as a binary relation, we do get the following analogue of Corollary 5.4.3.

Theorem 5.6.7. *The predicate*

$$\mathbf{ISO}(f,g) =_{\text{def}} (\mathfrak{M}_f \cong \mathfrak{M}_g) \tag{5.73}$$

is $\utilde{\Sigma}^1_1$-complete.

Proof. Let $\utilde{\mathcal{P}}(f)$ be a $\utilde{\Sigma}^1_1$ predicate. Then $\utilde{\mathcal{P}}(f) = \mathcal{P}(f,g)$ where $\mathcal{P}$ is Σ^1_1 and $g \in \omega^\omega$. For each $f \in \omega^\omega$, let $\mathfrak{A}_f$ be as in Proposition 5.6.5 and let $\mathfrak{B}_f \cong \omega_1^{f\oplus g}(1+\eta)$. Then the pair $(\mathfrak{A}_f, \mathfrak{B}_f)$ is uniformly computable in f and g, and clearly

$$\mathfrak{A}_f \cong \mathfrak{B}_f \quad \Leftrightarrow \quad \utilde{\mathcal{P}}(f) \tag{5.74}$$

$\square$

Chapter 6

Completeness Results for Hyperarithmetic Properties

6.1 Introduction

In Chapter 5, one of our main reasons for proving completeness results for analytic properties was to show that they could not be defined by a computable infinitary sentence. However, there are many model theoretic properties, like homogeneity, which are definable by a computable infinitary sentence. For these properties, we are interested in exactly how complicated this defining sentence must be.

Again, we can answer this question with completeness results. For example, if homogeneity is $\Pi^0_{\omega+2}$-complete, then the sentence in (5.2) defining this property is, in some sense, the best possible. Therefore, in this chapter we will prove several completeness results for various hyperarithmetic properties.

For the most part, the properties that we investigate in this chapter will be nonarithmetic. Therefore, the fact that our enumeration of computable models $\mathfrak{M}_e$ from Chapter 5 is only Δ^0_2 will not be a problem.

6.2 Elementary equivalence

One of the central themes of model theory is elementary equivalence, and many hyperarithmetic properties are defined using this property. For example, the property of homogeneity in Definition 5.1.2 is a property that gives us a way interpret a finite number of constants in the structure while still remaining elementary equivalent to the standard interpretation. Therefore, determining the complexity of elementary equivalence can help us determine the complexity of many other model theoretic properties.

There are a few ways that we can approach elementary equivalence. The most straightforward is to consider the complexity of $\equiv$ as a binary relation, by defining the predicate

$$\text{EQUIV}(e, i) =_{\text{def}} (\mathfrak{M}_e \equiv \mathfrak{M}_i) \tag{6.1}$$

We can also look at elementary equivalence relative to a fixed structure $\mathfrak{B}$.

$$\text{EQUIV}_{\mathfrak{B}}(e) =_{\text{def}} (\mathfrak{M}_e \equiv \mathfrak{B}) \tag{6.2}$$

By definition, for any structures $\mathfrak{A}$ and $\mathfrak{B}$,

$$\mathfrak{A} \equiv \mathfrak{B} \quad \Leftrightarrow \quad \bigwedge_{n \in \omega} \left(\forall \varphi \in \exists_n \left(\mathfrak{A} \models \varphi \leftrightarrow \mathfrak{B} \models \varphi\right)\right) \tag{6.3}$$

Therefore, it is clear that EQUIV is at most Π^0_ω. On the other hand, $\text{EQUIV}_{\mathfrak{B}}$ depends on the complexity of $\mathfrak{B}$, and may not even be hyperarithmetical. However, if $\mathfrak{B}$ is computable, it is at most as complicated as EQUIV. Therefore, our goal of this section is to prove that there is a computable model $\mathfrak{B}$ for which $\text{EQUIV}_{\mathfrak{B}}$ is Π^0_ω-complete.

This will require us to look at some computable structures first explored by Hirschfeldt [24]. In his thesis, Hirschfeldt explored the degree spectrum of definable relations on computable models. In particular, for each n, he constructed a computable structure with an invariant relation whose degree spectrum is exactly the Σ^0_n degrees.

His construction works because the elements of the invariant relation are distinguishable by a Σ^0_n formula, but not by any formula of lesser quantifier complexity. As a result, this construction can be used to build, for each $n \geq 1$, structures $\mathfrak{A}$, $\mathfrak{B}$ which agree on $\exists_{n-1}$ sentences, but not on $\exists_n$ sentences.

These structures will prove useful for investigating several properties, so we will extend them here into the computable ordinals. These structures are all R-trees; that is, they are connected, acyclic graphs with respect to the binary relation R, in which every element has at most one predecessor. Since there will be no confusion between these R-trees and the trees from Chapter 5, for the rest of this chapter we will say trees when we mean R-trees.

Throughout this chapter, we will use the downward-growth metaphor for our trees. We will refer to the unique element with no predecessor as the root. The unique predecessor of any element is the parent, and the successors are the children. The depth of a node a in a tree $\mathcal{T}$, which we denote $\text{depth}_{\mathcal{T}}(a)$, is its distance from the root.

Our trees are defined by induction on the computable ordinals. In the definition, trees associated with lesser ordinals will be a special type of subtree of those of higher ordinals.

Definition 6.2.1. Let $\mathcal{T}$ be an tree. A tree $\mathcal{S}$ is a *limb* of $\mathcal{T}$ if $\mathcal{S} \subseteq \mathcal{T}$ and

$$\forall x \in \mathcal{S}\, \forall y \in \mathcal{T}\, (x \; R \; y \rightarrow y \in \mathcal{S}) \tag{6.4}$$

If $a \in \mathcal{S}$ is the parent to the root of $\mathcal{S}$, we say that $\mathcal{S}$ is a *limb attached to a in $\mathcal{T}$*. For convenience, we often say that $\mathcal{S}$ is a limb attached to a so long as it is isomorphic to a limb attached to a. If $\mathcal{S}$ is a limb of $\mathcal{T}$, the *depth* of $\mathcal{S}$ in $\mathcal{T}$ is the depth of the root node of $\mathcal{S}$.

In other words, a limb of $\mathcal{T}$ with root a is the maximal subtree of $\mathcal{T}$ with root a. This will allow us to inductively define our trees by unambiguously specifying the subtrees attached to the root node.

For limit ordinals greater than ω, the definition of our trees will depend on our choice of notation for our computable ordinals. Throughout, we will assume a fixed set $\widetilde{O}$ of unique notations for computable ordinals, and we will identify each α with its notation $a \in \widetilde{O}$. Furthermore, if α is a limit ordinal, and $\{e\}$ is the fundamental sequence for α in $\widetilde{O}$, we will write α_n for $\{e\}(n)$. Furthermore, because the trees $\mathcal{A}_\delta$, $\mathcal{E}_\delta$ below are undefined when δ is a limit ordinal, we will require that our fundamental sequences $\langle \alpha_n \rangle_{n \in \omega}$ consist only of successor ordinals.

Definition 6.2.2. We define the *back-and-forth trees* by induction on α.

(a) $\mathcal{A}_1$ consists of a single node.

(b) $\mathcal{E}_1$ consists of a root node, to which are attached infinitely many nodes with no children. This tree is illustrated in Figure 6.1.

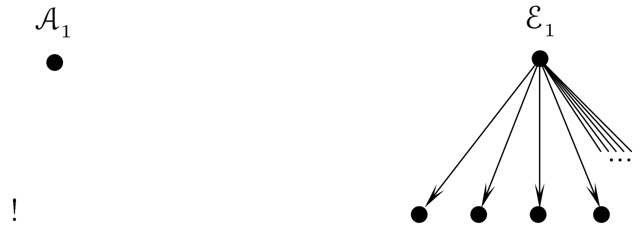

Figure 6.1: The base back-and-forth trees

(c) For any successor ordinal $\alpha+1$, $\mathcal{A}_{\alpha+2}$ consists of a root node, with infinitely many $\mathcal{E}_\alpha$ limbs attached to this root. This tree is illustrated in Figure 6.2.

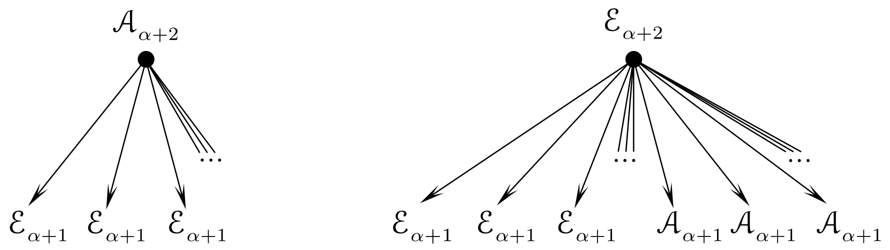

Figure 6.2: The successor back-and-forth trees

(d) For any successor ordinal $\alpha+1$, $\mathcal{E}_{\alpha+2}$ consists of a root node, with infinitely many $\mathcal{A}_\alpha$ limbs and infinitely $\mathcal{E}_\alpha$ limbs attached this root. This tree is also illustrated in Figure 6.2.

(e) For any limit ordinal γ and $k \in \omega$, $\mathcal{L}^\gamma_k$ consists of a root node with exactly one $\mathcal{A}_{\gamma_n}$ limb attached to this root for $n \leq k$, and exactly one $\mathcal{E}_{\gamma_n}$ limb attached to this root for $n > k$. This is well-defined, since our fundamental sequences consist only of successor ordinals. For convenience, we often denote $\mathcal{L}^\omega_n$ as $\mathcal{L}_n$. This tree is illustrated in Figure 6.3.

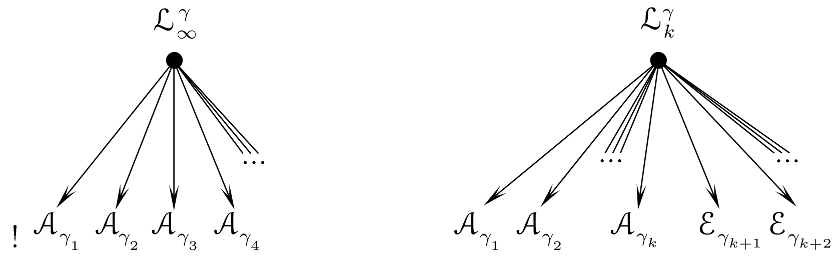

Figure 6.3: The limit back-and-forth trees

(f) For any limit ordinal γ, $\mathcal{L}^\gamma_\infty$ consists of a root node with exactly one $\mathcal{A}_{\gamma_n}$ limb attached to this root for each $n \in \omega$. Again, for convenience, we often denote $\mathcal{L}^\omega_\infty$ as $\mathcal{L}_\infty$. This tree is also illustrated in Figure 6.3.

(g) For any limit ordinal γ, $\mathcal{A}_{\gamma+1}$ consists of a root node with infinitely many $\mathcal{L}^\gamma_n$ limbs attached to this root for each $n \in \omega$. This tree is also illustrated in Figure 6.4.

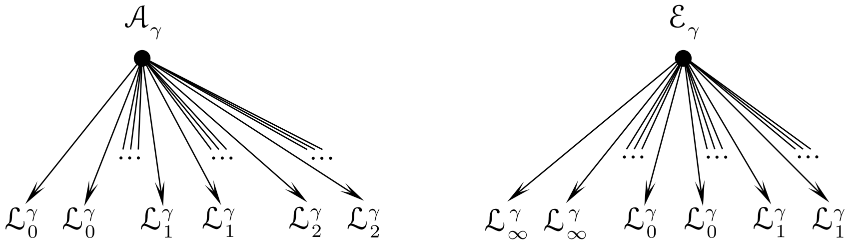

Figure 6.4: The transition back-and-forth trees

(h) For any limit ordinal γ, $\mathcal{E}_{\gamma+1}$ consists of a root node with infinitely many $\mathcal{L}^\gamma_n$ limbs attached to this root for each $n \in \omega \cup \{\infty\}$. This tree is illustrated in Figure 6.4.

Note that all of these trees are computably presentable by effective transfinite recursion on the computable ordinals. The following proposition demonstrates why these trees are useful for showing the completeness of model theoretic properties.

Proposition 6.2.3 (Modification of Hirschfeldt [24]). *Let $\mathcal{P}(n)$ be Σ^0_α.*

(a) If α is a successor ordinal, there is sequence of trees $\mathcal{T}_n$, uniformly computable from a Σ^0_α index for P such that for all n,

$$\mathcal{T}_n \cong \begin{cases} \mathcal{E}_\alpha & \textit{if } \mathcal{P}(n) \\ \mathcal{A}_\alpha & \textit{otherwise} \end{cases} \tag{6.5}$$

(b) If α is a limit ordinal, there is sequence of trees $\mathcal{T}_n$, uniformly computable from a Σ^0_α index for P such that for all n,

$$\mathcal{T}_n \cong \begin{cases} \mathcal{L}^\alpha_\infty & \textit{if } \neg\mathcal{P}(n) \\ \mathcal{L}^\alpha_k \textit{ for some } k & \textit{otherwise} \end{cases} \tag{6.6}$$

Proof. We proceed by effective transfinite recursion.

Case $\alpha = 1$: Then $\mathcal{P}(n) = \exists x\, \mathcal{Q}(x, n)$ for some computable predicate $\mathcal{Q}$. We enumerate a root element into $\mathcal{T}_n$. So long as no witness appears for $\mathcal{Q}(x, n)$, we do not add any children. Once a witness appears, we add infinitely many children to $\mathcal{T}_n$. We can do this while keeping $\mathcal{T}_n$ computable by ensuring that the children are all elements greater than our witness for $\mathcal{Q}(x, n)$.

Case $\alpha = \beta + 1$**, with** β **a successor ordinal**: Then $\mathcal{P}(n) = \exists x\, \mathcal{Q}(x, n)$ for some Π^0_β predicate $\mathcal{Q}(x, n)$. It is clear that we can go uniformly from a Σ^0_α index for $\mathcal{P}$ to a Σ^0_β index for $\mathcal{Q}$. So we can apply our induction hypothesis.

As β is a successor ordinal, we have a computable sequence of trees $\mathcal{U}_{x,n}$ satisfying (6.5). For each n, let $\mathcal{T}_n$ be an tree with infinitely many $\mathcal{U}_{x,n}$ limbs attached to the root for each $x \in \omega$, and infinitely many $\mathcal{E}_\beta$ limbs. As each $\mathcal{U}_{x,n}$ is either $\mathcal{E}_\beta$ or $\mathcal{A}_\beta$, $\mathcal{T}_n$ is either $\mathcal{E}_\alpha$ or $\mathcal{A}_\alpha$. Furthermore, $\mathcal{T}_n$ is only $\mathcal{E}_\alpha$ when at least one of the $\mathcal{U}_{x,n}$ is $\mathcal{A}_\beta$, or equivalently, $\exists x\, \mathcal{Q}(x, n)$.

Case α **is a limit ordinal**: By padding, we can assume with out loss of generality that $\mathcal{P}(n)$ is a computable disjunction of $\mathcal{Q}_k(n) \in \Sigma_{\alpha_k}$, and that we can go uniformly from a Σ^0_α index for $\mathcal{P}$ to $\Sigma^0_{\alpha_k}$ indices for the $\mathcal{Q}_k$. By induction, we have a computable sequence $\mathcal{U}_{m,n}$ satisfying (6.5) for the predicate

$$\mathcal{C}(m, n) =_{\text{def}} \bigvee_{k \le m} \mathcal{Q}_k(n) \tag{6.7}$$

Let $\mathcal{T}_n$ be the tree whose limbs attached to the root are exactly one copy of $\mathcal{U}_{m,n}$ for each $m \in \omega$. Then $\mathcal{T}_n$ is $\mathcal{L}^\alpha_n$ for some $n \in \omega \cup \{\infty\}$. Moreover, $\mathcal{T}_n$ is $\mathcal{L}^\alpha_\infty$ exactly when $\mathcal{P}(n)$ fails. So the trees $\mathcal{T}_n$ satisfy (6.6).

Case $\alpha = \beta + 1$**, with** β **a limit ordinal**: Again we write $\mathcal{P}(n) = \exists x\, \mathcal{Q}(x, n)$ for some Π^0_β predicate $\mathcal{Q}(x, n)$. By induction, there is a computable sequence of trees $\mathcal{S}_{x,n}$ satisfying (6.6) for $\mathcal{Q}$. Let $\mathcal{T}_n$ be a tree with infinitely $\mathcal{L}^\alpha_m$ limbs attached to the root for each $m \in \omega$, and infinitely many $\mathcal{S}_{x,n}$ limbs attached to the root for each $x \in \omega$. It is clear that the trees $\mathcal{T}_n$ satisfy (6.5). ∎

By appropriately padding the indices of our computable structures, this result gives us several 1-reductions to the collection of back-and-forth trees. Therefore, we can calculate lower bounds for the complexity of several model theoretic properties by determining which back-and-forth trees have these properties. In particular, we can use the back-and-forth trees to get the following result.

Theorem 6.2.4. *There is a computable structure* $\mathfrak{B}$ *such that*

$$(\Sigma^0_\omega, \Pi^0_\omega) \leq_{1:1} (\overline{\mathrm{EQUIV}_{\mathfrak{B}}}, \mathrm{ISOM}_{\mathfrak{B}}) \tag{6.8}$$

Proof. We need only show that $\mathcal{L}_\infty$ is not elementarily equivalent to any of the trees $\mathcal{L}_n$. If this is true, we can just let $\mathfrak{B} = \mathcal{L}_\infty$ and use the sequence $\mathcal{S}_n$ from Proposition 6.2.3 for our reduction. First, we need the following lemmas.

Lemma 6.2.5. *For each $k \in \omega$, there is an $\exists_k$ formula $\varphi_{\mathcal{E}_k}$ such that for any tree $\mathcal{T}$ and any back-and-forth tree $\mathcal{S}$ which is a limb with root $a \in \mathcal{T}$,*

$$\mathcal{T} \models \begin{cases} \varphi_{\mathcal{E}_k}(a) & \mathcal{S} \cong \mathcal{E}_k \\ \neg\varphi_{\mathcal{E}_k}(a) & \textit{otherwise} \end{cases} \tag{6.9}$$

Proof. By definition, $\mathcal{A}_{k+1}$ is an tree whose limbs attached to the root node are all copies of $\mathcal{E}_k$. Similarly for $\mathcal{E}_{k+1}$. Hence we can define $\varphi_{\mathcal{E}_k}(x)$ by induction on k.

$$\varphi_{\mathcal{E}_1} =_{\mathrm{def}} \exists y\,(x\ R\ y) \tag{6.10}$$

and

$$\varphi_{\mathcal{E}_{k+1}} =_{\mathrm{def}} \exists y\,\big((x\ R\ y) \wedge \neg\varphi_{\mathcal{E}_k}(y)\big) \tag{6.11}$$

It is clear that this definition works. ■

Lemma 6.2.6. *For each $m \in \omega$, there is a $\exists_{m+1}$ formula $\varphi_{\mathcal{L}_m}(x)$, such that for any tree $\mathcal{T}$ and any limb $\mathcal{L}_k$ with root $a \in \mathcal{T}$,*

$$\mathcal{T} \models \begin{cases} \varphi_{\mathcal{L}_m}(a) & \textit{If } k = m \\ \neg\varphi_{\mathcal{L}_m}(a) & \textit{otherwise} \end{cases} \tag{6.12}$$

Proof. We define the formula

$$\psi^{\mathrm{len}}_m(x) = \exists y\,(x\ R^m\ y) \wedge \forall z\,(\neg x\ R^{m+1}\ y) \tag{6.13}$$

where $(x\ R^n\ y)$ is the formula

$$x\ R^n\ y =_{\mathrm{def}} \exists z_0, \ldots, z_n \left((x = z_0) \wedge (y = z_n) \wedge \bigwedge_{i<n} (y_i\ R\ y_{i+1})\right) \tag{6.14}$$

saying that y is reachable from x in n steps. This $\exists_2$ formula says that the limb with root x has at least one path of length m, and that all paths are no more than length m.

Then for any $(a\ R\ b)$, $\psi_m^{\text{len}}(b)$ holds if and only if b is the root of either the $\mathcal{E}_m$ or the $\mathcal{A}_m$ limb of $\mathcal{L}_k$, as appropriate. So we let

$$\begin{aligned}\varphi_{\mathcal{L}_m}(x) =_{\text{def}} \exists y, z \big((x\ R\ y) \wedge \psi_m^{\text{len}}(y) \wedge \neg\varphi_{\mathcal{E}_m}(y)\big) \wedge \\ \big((x\ R\ z) \wedge \psi_{m+1}^{\text{len}}(z) \wedge \varphi_{\mathcal{E}_{m+1}}(z)\big)\end{aligned} \tag{6.15}$$

It is clear that this $\exists_{m+1}$ formula works from the definition of $\mathcal{L}_k$ and Lemma 6.2.5 above. ■

Given these lemmmas, take any $m \in \omega$, and let φ be the sentence

$$\varphi =_{\text{def}} \forall x \left(\forall y \left(\neg y\ R\ x\right) \to \varphi_{\mathcal{L}_m}(x)\right) \tag{6.16}$$

Then, by Lemma 6.2.6, $\mathcal{L}_m \models \varphi$ and $\mathcal{L}_\infty \not\models \varphi$. Hence $\mathcal{L}_\infty \not\equiv \mathcal{L}_m$. □

Corollary 6.2.7. EQUIV *is* Π^0_ω*-complete.*

6.3 Equivalence for infinitary sentences

Elementary equivalence is restricted to equivalence over first order sentences. We can extend this notion to Σ_α sentences.

Definition 6.3.1. For any two structures $\mathfrak{A}$, $\mathfrak{B}$, we say that $\mathfrak{A}$ and $\mathfrak{B}$ are Σ_α-*equivalent*, which we write as $\mathfrak{A} \equiv^\alpha \mathfrak{B}$ if for all Σ_α sentences $\varphi \in \mathcal{L}_{\omega_1,\omega}$

$$\mathfrak{A} \models \varphi \quad \Leftrightarrow \quad \mathfrak{B} \models \varphi \tag{6.17}$$

Note that, by the following proposition, $\equiv^\omega$ implies $\equiv$.

Proposition 6.3.2. *For any limit ordinal* γ*,* $\mathfrak{A} \equiv^\gamma \mathfrak{B}$ *if and only if* $\mathfrak{A} \equiv^\alpha \mathfrak{B}$ *for all* $\alpha < \gamma$*.*

Proof. The only nontrivial case is $(\Leftarrow)$. Let $\varphi \in \mathcal{L}_{\omega_1,\omega}$ be a Σ_γ sentence. Then $\neg\varphi = \bigwedge_{n\in\omega} \psi_n$ where each ψ_n is an Σ_α sentence for some $\alpha < \gamma$. Hence if $\mathfrak{A} \equiv^\alpha \mathfrak{B}$ for all $\alpha < \gamma$

$$\mathfrak{A} \not\models \varphi \Leftrightarrow \forall n\,(\mathfrak{A} \models \psi_n) \Leftrightarrow \forall n\,(\mathfrak{B} \models \psi_n) \Leftrightarrow \mathfrak{B} \not\models \varphi \tag{6.18}$$

□

However, in general, it is not the case that $\equiv^\omega$ is the same as $\equiv$. For all $n \geq 1$, the Σ^0_n sentences are infinitary. For example the following is a computable Π^0_2 sentence.

$$\varphi^{\text{trans}} =_{\text{def}} \forall x, y \bigvee_{n\in\omega} (x\ R^n\ y) \tag{6.19}$$

For any structure $\mathfrak{A}$, $\mathfrak{A} \models \varphi^{\text{trans}}$ if and only if $\mathfrak{A}$ is a connected graph. By a simple compactness argument, there are graphs $\mathfrak{A} \equiv \mathfrak{B}$ such that $\mathfrak{A}$ is connected while $\mathfrak{B}$ is not.

Because of the similarity of $\equiv^\omega$ to $\equiv$, we are interested in the complexity of this predicate. However, we cannot get a reasonable upper bound on the complexity of $\equiv^\omega$. Our upper bound on the complexity of $\equiv$ given in (6.3) depended on the fact that we could computably enumerate all of the first order sentences of our language. We cannot do this with all infinitary Σ_α sentences.

However, we can enumerate the computable Σ^0_α sentences. Therefore, we can replace $\equiv^\omega$ with $\equiv^\omega_c$, which requires only that (6.17) holds for all computable Σ^0_α sentences φ. Hence we define the following predicate

$$\text{EQUIV}^\alpha(e,i) =_{\text{def}} (\mathfrak{M}_e \equiv^\alpha_c \mathfrak{M}_i) \tag{6.20}$$

and similarly the predicate

$$\text{EQUIV}^\alpha_{\mathfrak{B}}(e) =_{\text{def}} (\mathfrak{M}_e \equiv^\alpha \mathfrak{B}) \tag{6.21}$$

If we let $\{\, \phi^\alpha_k \,\}_{k\in\omega}$ be a computable enumeration of the computable Σ_α sentences, we have that for any structures $\mathfrak{A}$ and $\mathfrak{B}$,

$$\mathfrak{A} \equiv^\alpha \mathfrak{B} \quad \Leftrightarrow \quad \bigwedge_{n\in\omega} (\mathfrak{A} \models \phi^\alpha_k \leftrightarrow \mathfrak{B} \models \phi^\alpha_k)) \tag{6.22}$$

As a result, EQUIV^α is at most $\Pi^0_{\alpha+1}$ when α is a successor ordinal, and at most Π^0_α when α is a limit ordinal. Our goal of this section is to show that these calculations are lower bounds as well.

To do this, we will extend Lemma 6.2.5 and Lemma 6.2.6 to infinitary formulae for arbitrary back-and-forth trees.

Lemma 6.3.3. *Let $\alpha = \beta + 1 > \omega$ be a computable successor ordinal. There is a computable Π^0_β formula $\psi^{\text{rank}}_\alpha$, such that for any tree $\mathcal{T}$ and any back-and-forth tree $\mathcal{S}$ that is a limb with root $a \in \mathcal{T}$,*

$$\mathcal{T} \models \begin{cases} \psi^{\text{rank}}_\alpha(a) & \mathcal{S} \cong \mathcal{E}_\alpha \text{ or } \mathcal{A}_\alpha. \\ \neg\psi^{\text{rank}}_\alpha(a) & \text{otherwise.} \end{cases} \tag{6.23}$$

Proof. We proceed by effective transfinite recursion on $\beta \geq \omega$.

Case $\beta = \omega$: Let ψ^{len}_m be the sentence from (6.13). We define

$$\psi^{\text{rank}}_{\omega+1}(x) =_{\text{def}} \forall y, \Big(x \; R \; y \rightarrow \bigwedge_{n\in\omega} \exists! z \left((y \; R \; z) \wedge \psi^{\text{len}}_n(z)\right)\Big) \tag{6.24}$$

Since the formulae ψ^{len}_n are all first order formulae, this is a computable Π_ω formula.

Take any $\mathcal{T}$ with $a \in \mathcal{T}$ where the limb $\mathcal{S}$ with root a is a back-and-forth tree. Then $\mathcal{S}$ is either $\mathcal{E}_{\omega+1}$ or $\mathcal{A}_{\omega+1}$ if and only if every limb attached to a is isomorphic to $\mathcal{L}_n$ for some $n \in \omega$. Since $\mathcal{S}$ is a back-and-forth tree, this is true if and only if every child of a has exactly one limb of the form $\mathcal{E}_n$ or $\mathcal{A}_n$ for each $n \in \omega$. Since $\mathcal{E}_n$ and $\mathcal{A}_n$, $n \in \omega$ are the only back-and-forth trees with paths of bounded length, it is clear from the discussion in the proof of Lemma 6.2.6 that $\psi^{\text{rank}}_{\omega+1}$ satsifies (6.23).

Case $\beta \geq \omega$ is a successor ordinal: We define

$$\psi^{\text{rank}}_{\alpha}(x) =_{\text{def}} \forall y\,(x \; R \; y \to \psi^{\text{rank}}_{\beta}(y)) \tag{6.25}$$

By induction, this is a computable Π_β formula.

Take any $\mathcal{T}$ with $a \in \mathcal{T}$ where the limb $\mathcal{S}$ with root a is a back-and-forth tree. Then $\mathcal{S}$ is either $\mathcal{E}_\alpha$ or $\mathcal{A}_\alpha$ if and only if all of limbs attached to a are isomorphic to either $\mathcal{E}_\beta$ or $\mathcal{A}_\beta$. Therefore, this formula satisfies (6.23) by induction.

Case β is a limit ordinal: We define

$$\psi^{\text{rank}}_{\alpha}(x) =_{\text{def}} \forall y, \Big(x \; R \; y \to \bigwedge_{n\in\omega} \exists! z \left((y \; R \; z) \wedge \psi^{\text{rank}}_{\beta_n}(y)\right)\Big) \tag{6.26}$$

Since the formulae $\psi^{\text{rank}}_{\beta_n}$ are all computable $\Pi_{\beta_n - 1}$ formulae, this is a computable Π_β formula.

Take any $\mathcal{T}$ with $a \in \mathcal{T}$ where the limb $\mathcal{S}$ with root a is a back-and-forth tree. Then $\mathcal{S}$ is either $\mathcal{E}_\alpha$ or $\mathcal{A}_\alpha$ if and only if every limb attached to a is isomorphic to $\mathcal{L}^\beta_n$ for some $n \in \omega$. Since $\mathcal{S}$ is a back-and-forth tree, this is true if and only if every child of a has exactly one limb of the form $\mathcal{E}_{\beta_n}$ or $\mathcal{A}_{\beta_n}$ for each $n \in \omega$. Therefore, this formula satisfies (6.23) by induction. □

Lemma 6.3.4. *Let α be a computable ordinal.*

(a) If α is a successor ordinal, there is a Σ_α formula $\varphi_{\mathcal{E}_\alpha}(x)$, uniformly computable in α, such that for any tree $\mathcal{T}$ and any back-and-forth tree $\mathcal{S}$ that is a limb with root $a \in \mathcal{T}$,

$$\mathcal{T} \models \begin{cases} \varphi_{\mathcal{E}_\alpha}(a) & \mathcal{S} \cong \mathcal{E}_\alpha \\ \neg\varphi_{\mathcal{E}_\alpha}(a) & \textit{otherwise} \end{cases} \tag{6.27}$$

(b) If α a limit ordinal, then for all $m \in \omega$ there is a $\Sigma_{\alpha_{m+1}}$ formula $\varphi_{\mathcal{L}^\alpha_m}(x)$, uniformly computable in α, such that for any tree $\mathcal{T}$ and any back-and-forth tree $\mathcal{S}$ that is a limb with root $a \in \mathcal{T}$,

$$\mathcal{T} \models \begin{cases} \varphi_{\mathcal{L}^\alpha_m}(a) & \mathcal{S} \cong \mathcal{L}^\alpha_m \\ \neg\varphi_{\mathcal{L}^\alpha_m}(a) & \textit{otherwise} \end{cases} \tag{6.28}$$

Proof. As usual, we proceed by effective transfinite recursion on $\alpha \geq \omega$.

Case $\alpha = k \in \omega$: The formula $\varphi_{\mathcal{E}_k} m$ from Lemma 6.2.5 is sufficient.

Case $\alpha = \omega$: The formula $\varphi_{\mathcal{L}_m^\omega} =_{\text{def}} \varphi_{\mathcal{L}_m}$ is sufficient by Lemma 6.2.6.

Case $\alpha = \beta + 1$**,** β **a limit ordinal**: We define

$$\varphi_{\mathcal{E}_\alpha}(x) =_{\text{def}} \exists y \left((x \; R \; y) \wedge \bigwedge_{n \in \omega} \neg \varphi_{\mathcal{L}_n^\alpha}(y) \right) \tag{6.29}$$

By induction, this is a computable $\Sigma^0_{\beta+1}$ formula. It is clear that this formula satisfies (6.27) by induction on (6.28).

Case $\alpha = \beta + 1$**,** β **a successor ordinal**: As in Lemma 6.2.5, we define

$$\varphi_{\mathcal{E}_\alpha}(x) =_{\text{def}} \exists y \left((x \; R \; y) \wedge \neg \varphi_{\mathcal{E}_\beta}(x)(y) \right) \tag{6.30}$$

By induction, this is a computable $\Sigma^0_{\beta+1}$ formula satisfying (6.27).

Case α **a limit ordinal**: As in Lemma 6.2.6, we define

$$\begin{aligned} \varphi_{\mathcal{L}_m^\alpha}(x) =_{\text{def}} \exists y, z \left((x \; R \; y) \wedge \psi^{\text{rank}}_{\alpha_m}(y) \wedge \neg \varphi_{\mathcal{E}_{\alpha_m}}(y) \right) \wedge \\ \left((x \; R \; y) \wedge \psi^{\text{rank}}_{\alpha_{m+1}}(y) \wedge \varphi_{\mathcal{E}_{\alpha_{m+1}}}(y) \right) \end{aligned} \tag{6.31}$$

By induction, this is a computable $\Sigma^0_{\alpha_{m+1}}$ formula satisfying (6.28). □

Given these two lemmas, we can now prove the following analogue of Theorem 6.2.4.

Theorem 6.3.5. *Let $\alpha \geq 1$ be a computable ordinal.*

(a) If α is a successor ordinal, there is a computable structure $\mathfrak{B}$ such that

$$(\Sigma_{\alpha+1}, \Pi_{\alpha+1}) \leq_{1:1} (\overline{\text{EQUIV}^\alpha_{\mathfrak{B}}}, \text{ISOM}_{\mathfrak{B}}) \tag{6.32}$$

(b) If α is a limit ordinal, there is a computable structure $\mathfrak{B}$ such that

$$(\Sigma_\alpha, \Pi_\alpha) \leq_{1:1} (\overline{\text{EQUIV}^\alpha_{\mathfrak{B}}}, \text{ISOM}_{\mathfrak{B}}) \tag{6.33}$$

Proof. The proof for claim (b) is exactly the same as Theorem 6.2.4; we just let $\mathfrak{B} = \mathcal{L}^\alpha_\infty$. So we need only prove claim (a).

Let $\mathcal{P}(n)$ be a $\Pi^0_{\alpha+1}$ predicate. Intuitively, we would like to set $\mathfrak{B} = \mathcal{A}_{\alpha+1}$. Then we have a computable sequence of trees $\mathcal{T}_n$ such that $\mathcal{T}_n \cong \mathcal{A}_{\alpha+1}$ if $P(n)$ holds and $\mathcal{T}_n \cong \mathcal{E}_{\alpha+1}$ otherwise. Unfortunately, this is not sufficient, as Lemma 6.3.4 only guarantees that $\mathcal{A}_{\alpha+1} \not\equiv^{\alpha+1} \mathcal{E}_{\alpha+1}$. In fact, as we shall see in section 6.8, $\mathcal{A}_{\alpha+1} \equiv^\alpha \mathcal{E}_{\alpha+1}$.

To get around this problem, we introduce the following collection of trees.

Definition 6.3.6. Take any $k \in \omega$. The *k-tagged intrepretation* of $\mathcal{A}_\alpha(\mathcal{E}_\alpha)$ is the structure $\mathfrak{A}^k_\alpha(\mathfrak{E}^k_\alpha)$ with the following properties.

(a) The structure $\mathfrak{A}^k_\alpha(\mathfrak{E}^k_\alpha)$ has a unique R-cycle of length $k+1$.

(b) A copy of $\mathcal{A}_\alpha(\mathcal{E}_\alpha)$ is attached to exactly one of the nodes in this unique R-cycle of length $k+1$. We call this node the *k-handle*.

(c) Every element of this structure is either a node in the R-cycle of length $k+1$, or a node in the copy of $\mathcal{A}_\alpha(\mathcal{E}_\alpha)$.

These structures are illustrated in Figure 6.5.

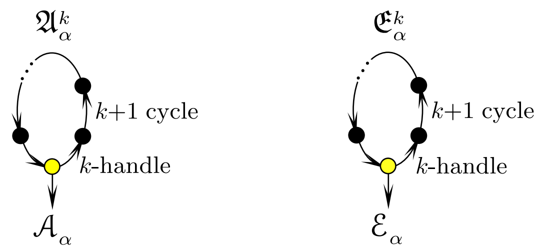

Figure 6.5: The k-tagged intrepretations $\mathfrak{A}^k_\alpha$ and $\mathfrak{E}^k_\alpha$

We let $\mathfrak{B}$ be the disjoint union of the structures $\mathfrak{A}^k_\alpha$, for $k \in \omega$. We let $\alpha = \beta+1$ and write $\mathcal{P}(n) = \forall k\, \mathcal{Q}(x, n)$ where $\mathcal{Q}$ is a Σ^0_β predicate. Using Proposition 6.2.3, we can uniformly construct structures $\mathfrak{D}_{k,n}$ such that

$$\mathfrak{D}_{k,n} \cong \begin{cases} \mathfrak{E}^k_\alpha & \text{If } \mathcal{Q}(k,n) \\ \mathfrak{A}^k_\alpha & \text{otherwise} \end{cases} \tag{6.34}$$

For each $n \in \omega$, we let $\mathfrak{D}_n$ be the disjoint union of the structures $\mathfrak{D}_{k,n}$, for $k \in \omega$. It is clear that $\mathfrak{D}_n \cong \mathfrak{B}$ whenever $\mathcal{P}(n)$ holds. We need only show that $\mathfrak{D}_n \not\equiv^\alpha \mathfrak{B}$ whenever $\exists k\, \neg\mathcal{Q}(x, n)$.

Let $k \in \omega$ be such that $\mathfrak{D}_{k,n} \cong \mathfrak{A}^k_\alpha$. Using the computable Σ_α formula $\varphi_{\mathcal{E}_\alpha}(y)$ from Lemma 6.3.4, we can define the sentence

$$\varphi =_{\mathrm{def}} \exists x, y\, \big((x\ R^{k+1}\ x) \wedge (x\ R\ y) \wedge \varphi_{\mathcal{E}_\alpha}(y)\big) \tag{6.35}$$

Since $\alpha \geq 1$, this is a Σ_α sentence. Even though these structures are not trees, it is clear from the proof of Lemma 6.3.4 that $\mathfrak{B} \models \varphi$ while $\mathfrak{D}_n \not\models \varphi$. $\square$

Corollary 6.3.7. *$\equiv^\alpha_c$ is $\Pi^0_{\alpha+1}$-complete whenever α is a successor ordinal, and Π^0_α-complete whenever α is a limit ordinal.*

6.4 Homogeneity

Intuitively, these back-and-forth trees should be helpful at analyzing the complexity of any property whose definition includes elementary equivalence. We have already mentioned that homogeneity is such a property. We saw in Chapter 5 that homogeneity is at most $\Pi^0_{\omega+2}$. Therefore, our goal of this section is to prove the following.

Theorem 6.4.1. HOM *is* $\Pi^0_{\omega+2}$*-complete.*

To prove this theorem, it is enough to show that $\mathcal{A}_{\omega+2}$ is homogeneous while $\mathcal{E}_{\omega+2}$ is not. In this case, Proposition 6.2.3 will give us our required 1-reduction.

The proof that $\mathcal{A}_{\omega+2}$ is homogeneous will use the formulae from Lemma 6.2.5 and Lemma 6.2.6. The proof that $\mathcal{E}_{\omega+2}$ is not homogeneous will require us to show that there are $a, b \in \mathcal{E}_{\omega+2}$ which are not in the same automorphism orbit, but for which $(\mathcal{E}_{\omega+2}, a) \equiv (\mathcal{E}_{\omega+2}, b)$. For this latter step, we will need a technique pioneered by Roland Fraïssé in the 1950s, now known as Ehrenfeucht-Fraïssé games. These games are superficially similar to those of Chapter 3. As before, we have two players player$_\forall$ and player$_\exists$. However, instead of choosing finite sets of sentences, the players choose elements from a pair of structures $\mathfrak{A}$ and $\mathfrak{B}$.

Let $n < \omega$, and $\mathfrak{A}$, $\mathfrak{B}$ any two structures of our language $\mathcal{L}$. At stages $k < n$, player$_\forall$ chooses one of the two structures, and an element from that structure; player$_\exists$ then chooses an element in the other structure. As before, both players are allowed to use the results of previous stages in determining their choice. At the end of this stage, the two players have chosen elements $a_k \in \mathfrak{A}$ and $b_k \in \mathfrak{B}$. This game goes on for n stages until we have sequences $\overline{a} = (a_k \,|\, k < n)$ and $\overline{b} = (b_k \,|\, k < n)$.

Definition 6.4.2. The game described above is the *Ehrenfeucht-Fraïssé game of length n* on $\mathfrak{A}$ and $\mathfrak{B}$, denoted $\mathrm{EF}_n(\mathfrak{A}, \mathfrak{B})$. The pair $(\overline{a}, \overline{b})$ is a *play* of this game. A play $(\overline{a}, \overline{b})$ is a *win for player$_\exists$* if $\overline{a} \mapsto \overline{b}$ induces an isomorphism from the substructure $\langle \overline{a} \rangle_{\mathfrak{A}}$ to the substructure $\langle \overline{b} \rangle_{\mathfrak{B}}$. Any play that is not a win for player$_\exists$ is a win for player$_\forall$.

As with the games in Chapter 3, we are interested when player$_\exists$ has a winning strategy for the game $\mathrm{EF}_n(\mathfrak{A}, \mathfrak{B})$.

Definition 6.4.3. We define a winning strategy for the game $\mathrm{EF}_n(\mathfrak{A}, \mathfrak{B})$ in the usual way. If player$_\exists$ has a winning strategy for this game, we write $\mathfrak{A} \approx_n \mathfrak{B}$.

Ehrenfeucht-Fraïssé games are a useful tool because they allow us to study equivalence with respect to sentences of finite quantifier rank.

Definition 6.4.4. For a formula $\varphi \in \mathcal{L}_{\omega_1,\omega}$, we define the *quantifier rank* of φ, which we denote $\mathrm{qr}(\varphi)$, by induction on the complexity of φ.

(a) If φ is atomic, then $\mathrm{qr}(\varphi) = 0$.

(b) If $\varphi = \neg\psi$, then $\mathrm{qr}(\varphi) = \mathrm{qr}(\psi)$

(c) If $\varphi = \bigwedge_\alpha \psi_\alpha$, then $\mathrm{qr}(\varphi) = \sup_\alpha(\mathrm{qr}(\psi_\alpha))$.

(d) If $\varphi = \exists x\, \psi(x)$, then $\mathrm{qr}(\varphi) = \mathrm{qr}(\psi) + 1$

For structures $\mathfrak{A}$ and $\mathfrak{B}$, we say that $\mathfrak{A} \equiv_n \mathfrak{B}$ if they satisfy exactly the same sentences of quantifier rank up to and including n.

It is clear that any first order sentence has finite quantifier rank. Therefore, we see from the following result that we can prove $\mathfrak{A} \equiv \mathfrak{B}$ by constructing a winning strategy for $\mathrm{EF}_n(\mathfrak{A}, \mathfrak{B})$ for each $n \in \omega$.

Theorem 6.4.5 (Fraïssé [12],Ehrenfeucht [10]). *Let $\mathfrak{A}$, $\mathfrak{B}$ be structures in a language with finite signature. For all $n \in \omega$, $\mathfrak{A} \equiv_n \mathfrak{B}$ if and only if $\mathfrak{A} \approx_n \mathfrak{B}$.*

With this in mind, we use Ehrenfeucht-Fraïssé games to prove the following sequence of lemmas.

Lemma 6.4.6. *For all $n \geq 1$, and $a \in \mathcal{A}_n$, there is a $b \in \mathcal{E}_n$ such that $(\mathcal{A}_n, a) \approx_{n-1} (\mathcal{E}_n, b)$.*

Proof. We proceed by induction on $n \in \omega$.

Case $n = 1$: Clearly a must be the single node of $\mathcal{A}_k$. Let b be the root node of $\mathcal{E}_k$. Then we have that

$$(\mathcal{A}_0, a) \vDash (\neg a \ R \ a) \quad \text{and} \quad (\mathcal{E}_0, b) \vDash (\neg b \ R \ b) \tag{6.36}$$

The substructures $\langle a \rangle_{\mathfrak{A}}$ and $\langle b \rangle_{\mathfrak{B}}$ each have a single element. Therefore they are isomorphic by (6.36).

Case $n = k + 1$: Take any $a \in \mathcal{A}_{k+1}$. If a is the root, let b be the root of $\mathcal{E}_{k+1}$. Otherwise, a is some element of an $\mathcal{E}_k$ limb attached to the root of $\mathcal{A}_{k+1}$. In this case we choose b from an $\mathcal{E}_k$ limb of $\mathcal{E}_{k+1}$ so that $(\mathcal{E}_k, a) \cong (\mathcal{E}_k, b)$. We need only show this choice of b works.

We will construct a winning strategy for player$_\exists$ in the Ehrenfeucht-Fraïssé game $\mathrm{EF}_{k+1}((\mathcal{A}_{k+1}, a), (\mathcal{E}_{k+1}, b))$. Our strategy will be as follows.

- If player$_\forall$ chooses the root node of either structure, we match that with the root node of the other structure.
- If player$_\forall$ chooses an element c for the first time in an $\mathcal{E}_k$ limb of either of the two structures, we choose an fresh $\mathcal{E}_k$ limb in the other structure, and say that these two limbs are *mated*. We also choose some $d \in \mathcal{E}_k$ such that $(\mathcal{E}_k, c) \cong (\mathcal{E}_k, d)$. We say that c, d are the *mating nodes* of this mated pair.
- If player$_\forall$ chooses an element of a mated pair of $\mathcal{E}_k$, we continue to realize the isomorphism guaranteed by the mating nodes.
- If player$_\forall$ chooses an element c for the first time in an $\mathcal{A}_k$ limb of $\mathcal{E}_{k+1}$, we choose an fresh $\mathcal{E}_k$ limb in $\mathcal{A}_{k+1}$, and say that these two children are *mated*. Using the inductive hypothesis, we choose some $d \in \mathcal{E}_k$ such that $(\mathcal{A}_k, c) \approx_{k-1} (\mathcal{E}_k, d)$, and call c, d the *mating nodes* as before.
- If player$_\forall$ chooses an element of a mated pair $\mathcal{A}_k$, $\mathcal{E}_k$, we continue to play the strategy for $(\mathcal{A}_k, c) \approx_{k-1} (\mathcal{A}_k, d)$ where c, d are the mating nodes.

When the game is done, we will have $k+1$ elements chosen from each of the structures $\mathcal{A}_{k+1}$, $\mathcal{A}_{k+1}$ and an assignment between them. The game is won if our construction restricted to the $k+1$ elements in each structure defines an isomorphism. The root of one tree, if chosen, is always mapped to the root of the other. So it is clear that we need only show that our construction defines an isomorphism between elements chosen in mated limbs.

The only questionable case is for a mated pair $\mathcal{A}_k$, $\mathcal{E}_k$. Note that our original a, b cannot appear in such a mated pair. So after our original mating nodes c, d, we can choose at most $k-1$ other elements from this mated pair. As we applied a winning strategy for a game of length $k-1$ between this mated pair, we are done. $\square$

Lemma 6.4.7. *For all $k < n \leq \infty$, $\mathcal{L}_k \approx_k \mathcal{L}_n$.*

Proof. Our winning strategy for player$_\exists$ will be as follows.

- If player$_\forall$ chooses the root node of either structure, we match that with the root node of the other structure.
- If player$_\forall$ chooses an element c for the first time in an $\mathcal{A}_m$ limb of either of the two structures, $m \leq k$, we mate it with the $\mathcal{A}_m$ limb of the other structure. We also choose some $d \in \mathcal{A}_k$ such that $(\mathcal{A}_k, c) \cong (\mathcal{A}_k, d)$, and define the c, d as mating nodes.
- If player$_\forall$ chooses an element c for the first time in an $\mathcal{E}_m$ limb of either of the two structures, $m > n$, we mate it with the $\mathcal{E}_m$ limb of the other structure. We also choose some $d \in \mathcal{E}_k$ to make mating nodes as usual.
- If player$_\forall$ chooses an element c for the first time in an $\mathcal{A}_m$ limb of $\mathcal{L}_k$, $k < m \leq n$, we mate it with the $\mathcal{E}_m$ limb in $\mathcal{L}_n$. We also choose some $d \in \mathcal{E}_m$ such that $(\mathcal{A}_m, c) \approx_{m-1} (\mathcal{E}_m, d)$, to define the mating nodes.
- If player$_\forall$ chooses an element c for the first time in an $\mathcal{E}_m$ limb of $\mathcal{L}_n$, $k < m \leq n$, we mate it with the $\mathcal{A}_m$ limb in $\mathcal{L}_k$. We also choose some $d \in \mathcal{A}_k$ such that $(\mathcal{E}_m, c) \approx_{m-2} (\mathcal{A}_m, d)$, to define the mating nodes. We can do this because $\mathcal{A}_m \approx_{m-1} \mathcal{E}_m$, so we just let d be the move following the choice of $c \in \mathcal{E}_m$ in this strategy.
- If player$_\forall$ chooses an element of a mated pair, we continue to play the strategy for that pair given by the mating nodes.

The proof that this strategy works is analogous to that of Lemma 6.4.6. $\square$

Lemma 6.4.8. $\mathcal{E}_{\omega+1} \equiv \mathcal{A}_{\omega+1}$.

Proof. By Theorem 6.4.5, we need only show that $\mathcal{E}_{\omega+1} \approx_n \mathcal{A}_{\omega+1}$, for all $n \in \omega$. Given $n \in \omega$, we define a winning strategy for player$_\exists$ in the game $\mathrm{EF}_n(\mathcal{E}_{\omega+1}, \mathcal{A}_{\omega+1})$ as follows.

- If player$_\forall$ chooses the root node of either structure, we match that with the root node of the other structure.
- If player$_\forall$ chooses an element c for the first time in an $\mathcal{L}_k$ limb of either of the two structures, $k \in \omega$, we mate it with the $\mathcal{L}_k$ limb of the other structure. We also choose some $d \in \mathcal{L}_k$ such that $(\mathcal{L}_k, c) \cong (\mathcal{L}_k, d)$, and define the c, d as the mating nodes.
- If player$_\forall$ chooses an element c for the first time in an $\mathcal{L}_\infty$ limb of $\mathcal{E}_{\omega+1}$, we mate it with the $\mathcal{E}_n$ limb in $\mathcal{A}_{\omega+1}$. We also choose some $d \in \mathcal{L}_{n+1}$ such that $(\mathcal{L}_\infty, c) \approx_{n-1} (\mathcal{L}_{n+1}, d)$, to define the mating nodes. We can do this by Lemma 6.4.7.
- If player$_\forall$ chooses an element of a mated pair, we continue to play the strategy for that pair given by the mating nodes.

□

These lemmas give us the first half of Theorem 6.4.1.

Proposition 6.4.9. *$\mathcal{E}_{\omega+2}$ is not homogeneous.*

Proof. Let a, b be the roots of some $\mathcal{A}_{\omega+1}$, $\mathcal{E}_{\omega+1}$ limbs of $\mathcal{E}_{\omega+2}$, respectively. Since the root nodes are definable by the formula

$$\varphi^{\mathrm{root}}(x) =_{\mathrm{def}} \forall y\, (\neg y \; R \; x) \tag{6.37}$$

we have that $(\mathcal{A}_{\omega+1}, a) \equiv (\mathcal{E}_{\omega+1}, b)$ by Lemma 6.4.8. We need only extend this to show that $(\mathcal{E}_{\omega+2}, a) \equiv (\mathcal{E}_{\omega+2}, b)$.

Take any $n \in \omega$. Our strategy for the game $\mathrm{EF}_n((\mathcal{E}_{\omega+2}, a), (\mathcal{E}_{\omega+2}, b))$ is the same as before. The root node is mapped to the root node. Every time player$_\forall$ chooses from a limb other than the ones containing a, b, we mate it to an isomorphic limb. Otherwise, we play the strategy for $(\mathcal{A}_{\omega+1}, a)$ and $(\mathcal{E}_{\omega+1}, b)$. It is clear that this strategy works. □

Therefore, to prove Theorem 6.4.1, it only remains for us to show that $\mathcal{A}_{\omega+2}$ is homogeneous. To do this, we need to show that when two sequences $\overline{a}$, $\overline{b}$ have the same type, we can construct an automorphism between them. Our automorphisms will be constructed by induction on the definition of the back-and-forth trees. To do this, we first need a way to match up the limbs of a back-and-forth tree.

Definition 6.4.10. Let $\mathcal{T}$ be a tree with $\overline{a} \in \mathcal{T}$. Let $\{\,\mathcal{T}_n\,\}_{n\in\omega}$ be an enumeration of the limbs attached to the root of $\mathcal{T}$. For each $n \in \omega$, we denote the *restriction of $\overline{a}$ to $\mathcal{T}_n$* as the sequence $\overline{a}|_n$.

Lemma 6.4.11. *Let $\mathcal{T}$ be a countable tree with a limb $\mathcal{U}$ such that $\mathcal{U}$ is a back-and-forth tree. Let $\mathcal{S}$ be a countable tree with a limb $\mathcal{W} \cong \mathcal{U}$ having the same depth as $\mathcal{U}$. Suppose that $\overline{a} \in \mathcal{U}$ and $\overline{b} \in \mathcal{W}$ are such that $(\mathcal{T}, \overline{a}) \equiv (\mathcal{S}, \overline{b})$. Finally let $\{\, \mathcal{U}_n \,\}_{n\in\omega}$ and $\{\, \mathcal{W}_n \,\}_{n\in\omega}$ be enumerations of the limbs attached to the roots of $\mathcal{U}$ and $\mathcal{W}$ respectively. Then there is a permutation $\pi : \omega \to \omega$ such that for all $i \leq |\overline{a}|$*

$$a_i \in \overline{a}|_n \quad \Leftrightarrow \quad b_i \in \overline{b}|_{\pi(n)} \tag{6.38}$$

Proof. Let d be the depth of $\mathcal{U}$ in $\mathcal{T}$. We define the formula

$$\chi_d^{\text{depth}}(x) =_{\text{def}} \exists x, y \left(\varphi^{\text{root}}(x) \wedge (x\ R^d\ y)\right) \tag{6.39}$$

As $\mathcal{W}$ has the same depth in $\mathcal{S}$ and $\mathcal{U}$ does in $\mathcal{S}$, for all $a_i \in \overline{a}$,

(6.40)

$$a_i \text{ is the root of } \mathcal{U} \Leftrightarrow \mathcal{T} \vDash \chi_d^{\text{depth}}(a_i) \Leftrightarrow \mathcal{S} \vDash \chi_d^{\text{depth}}(b_i) \Leftrightarrow b_i \text{ is the root of } \mathcal{W}$$

Fix some $n \in \omega$. There is some formula

$$\rho_{\overline{a}|_n}(w, \overline{x}) =_{\text{def}} \exists z \left(\chi_d^{\text{depth}}(z) \wedge (z\ R\ w) \wedge \bigwedge_{x_j \in \overline{x}} (w\ R^{k_j}\ x_j)\right) \tag{6.41}$$

such that

$$\mathcal{T} \vDash \exists! w \left(\rho_{\overline{a}|_n}(w, \overline{a}|_n)\right) \tag{6.42}$$

and this w is the root of $\mathcal{U}_i$. This formula says that the elements $\overline{a}|_n$ all sit together in a single limb of $\mathcal{U}$ with root w.

Let $\overline{d} \subseteq \overline{b}$ be the sequence of elements

$$b_i \in \overline{d} \quad \Leftrightarrow \quad a_i \in \overline{a}|_n \tag{6.43}$$

As $(\mathcal{T}, \overline{a}) \equiv (\mathcal{S}, \overline{b})$, we have that $\mathcal{S} \vDash \exists! w\, \rho_{\overline{a}|_n}(w, \overline{d})$. Therefore, since $\mathcal{W}$ has the same depth as $\mathcal{U}$, the constants $\overline{d}$ must all sit in a single limb $\mathcal{W}_m$ of $\mathcal{W}$. We set $\pi_0(n) = m$.

We extend the partial function π_0 to a function $\pi : \omega \to \omega$ as follows. For each $s \in \omega$, let $k \in \omega$ be the least element no in the domain of π_s, and $m \in \omega$ the least element not in the range of π_s. We define

$$\pi_{s+1}(n) = \begin{cases} \pi_s(n) & n \in \operatorname{dom}(\pi_s) \\ m & n = k \\ \textbf{Undefined} & \text{otherwise} \end{cases} \tag{6.44}$$

We let $\pi = \bigcup_s \pi_s$. It is clear that this function is onto. We need only show that it is one-one.

Fix $n \in \omega$. Let $\overline{d}$ be the sequence of elements corresponding to (6.43). It is enough to show that $\overline{d} = \overline{b}|_{\pi(n)}$. Let $\overline{c} \supseteq \overline{a}|_n$ be the sequence of elements such that

$$a_i \in \overline{c} \quad \Leftrightarrow \quad b_i \in \overline{b}|_{\pi(n)} \tag{6.45}$$

We define $\rho_{\overline{a}|_n}$ as in (6.41). As $\mathcal{T} \vDash \exists! w\, \rho_{\overline{b}|_n}(w, \overline{c})$, the constants $\overline{c}$ all sit in a single limb $\mathcal{U}_k$ of $\mathcal{U}$. Therefore, $\overline{c} = \overline{a}|_n$ by definition. So $\overline{d} = \overline{b}|_{\pi(n)}$ $\square$

This lemma allows us to compose automorphisms between limbs together to form an automorphism of the entire tree.

Lemma 6.4.12. *Let $\mathcal{T}$ be a countable tree with a limb $\mathcal{U}$ such that one of the following is true.*

(a) $\mathcal{U} \cong \mathcal{E}_k$ *or* $\mathcal{A}_k$, *for* $k \in \omega$, *or*

(b) $\mathcal{U} \cong \mathcal{L}_m$, *for* $m \in \omega \cup \{\infty\}$, *or*

(c) $\mathcal{U} \cong \mathcal{E}_{\omega+1}$ *or* $\mathcal{A}_{\omega+1}$, *or*

(d) $\mathcal{U} \cong \mathcal{A}_{\omega+2}$.

Let $\mathcal{S}$ be a countable tree with a limb $\mathcal{W} \cong \mathcal{U}$ having the same depth as $\mathcal{U}$. Suppose that $\overline{a} \in \mathcal{U}$ and $\overline{b} \in \mathcal{W}$ are such that $(\mathcal{T}, \overline{a}) \equiv (\mathcal{S}, \overline{b})$. Then there is an isomorphism $f : \mathcal{U} \to \mathcal{W}$ which takes $\overline{a}$ to $\overline{b}$.

Proof. We proceed by induction on the back-and-forth trees listed in (a) – (d).

Case $\mathcal{U} \cong \mathcal{A}_1$: In this case, $\mathcal{U}$ and $\mathcal{W}$ both consist only of a single point, so our claim is trivial.

Case $\mathcal{U} \cong \mathcal{E}_1$: Let d be the depth of $\mathcal{U}$ in $\mathcal{T}$. Then (6.40) of Lemma 6.4.11 still holds. As any element that is not the root of $\mathcal{E}_0$ can be mapped to any other element that is not the root, it is clear that there is an isomorphism between the two limbs taking $\overline{a}$ to $\overline{b}$.

Case $\mathcal{U} \cong \mathcal{A}_{k+1}$: Let d be the depth of $\mathcal{U}$ in $\mathcal{T}$. Again, we have that (6.40) holds. Let $\mathcal{U}_n$ be an enumeration of the limbs attached to the root of $\mathcal{U}$, and similarly $\mathcal{W}_n$ for $\mathcal{W}$. Let $\pi : \omega \to \omega$ be the permutation of Lemma 6.4.11. It is enough to show that, for each $n \in \omega$, there is a isomorphism $f_n : \mathcal{U}_n \to \mathcal{W}_{\sigma(n)}$ which preserves the constants $\overline{a}|_n$, $\overline{b}_n$. In this case, we can compose the f_n together to get our isomorphism f. As the limbs $\mathcal{U}_n$, $\mathcal{W}_{\sigma(n)}$ are all of the same type, namely $\mathcal{E}_k$, the existence of the isomorphisms f_n follows from our inductive hypothesis.

Case $\mathcal{U} \cong \mathcal{E}_{k+1}$: The proof for this case is analogous to that of the case $\mathcal{U} \cong \mathcal{A}_{k+1}$. The only difference is that the limbs $\mathcal{U}_n$ can be either $\mathcal{E}_k$ or $\mathcal{A}_k$. So we need to show that $\mathcal{U}_n \cong \mathcal{W}_{\sigma(n)}$ for all $n \in \omega$.

Suppose $\mathcal{U}_n \cong \mathcal{E}_k$. Define the formula

$$\delta(\overline{x}) =_{\mathrm{def}} \exists! w \left(\rho_{\overline{a}|_n}(w, \overline{x}) \wedge \varphi_{\mathcal{E}_k}(w)\right) \tag{6.46}$$

$\varphi_{\mathcal{E}_k}$ is as in Lemma 6.2.5. As the $w \in \mathcal{T}$ which witnesses $\rho_{\overline{a}|_n}(w, \overline{c})$ is the root of $\mathcal{U}_n$, $\mathcal{T} \models \delta(\overline{a}|_n)$. Thus $\mathcal{S} \models \delta(\overline{b}|_n)$ and so the root of $\mathcal{W}_{\sigma(n)}$ satisfies $\varphi_{\mathcal{E}_k}$. Thus $\mathcal{W}_{\sigma(n)} \cong \mathcal{E}_k$ by Lemma 6.2.5.

The proof for $\mathcal{U}_n \cong \mathcal{A}_k$ is analogous. We simply replace the formula $\varphi_{\mathcal{E}_k}$ in (6.46) with $\neg\varphi_{\mathcal{E}_k}$.

Case $\mathcal{U} \cong \mathcal{L}_m$: Note that the limbs attached to the root of $\mathcal{U}$ are of the form $\mathcal{A}_k$ or $\mathcal{E}_k$, for $j \in \omega$. So, by induction, we can carry out the proof for the case $\mathcal{U} \cong \mathcal{E}_{k+1}$, provided that we show $\mathcal{U}_n$ and $\mathcal{W}_{\sigma(n)}$ for all $n \in \omega$.

Suppose $\mathcal{U}_n \cong \mathcal{E}_k$ for some k. We define the formula

$$\delta(\overline{x}) =_{\text{def}} \exists! w \left(\rho_{\overline{a}|_n}(w, \overline{x}) \wedge \varphi_{\mathcal{E}_k}(w) \wedge \psi_k^{\text{len}}(w)\right) \tag{6.47}$$

where ψ_k^{len} is as in (6.13) of Lemma 6.2.6. Again, $\mathcal{T} \models \delta(\overline{a}|_n)$, and so $\mathcal{S} \models \delta(\overline{b}|_n)$. Thus the root of $\mathcal{W}_{\sigma(n)}$ satisfies both ψ_k^{len} and $\varphi_{\mathcal{E}_k}$. From Lemma 6.3.3, we see that $\mathcal{W}_{\sigma(n)}$ is either $\mathcal{E}_k$ or $\mathcal{A}_k$, and hence $\mathcal{W}_{\sigma(n)} \cong \mathcal{E}_k$ by Lemma 6.2.5.

The proof for $\mathcal{U}_n \cong \mathcal{A}_k$ is analogous. We simply replace the formula $\varphi_{\mathcal{E}_k}$ in (6.47) with $\neg\varphi_{\mathcal{E}_k}$.

Case $\mathcal{U} \cong \mathcal{E}_{\omega+1}$: Note that all the limbs attached to the root of $\mathcal{U}$ are of the form $\mathcal{L}_m$, for some $m \in \omega \cup \{\infty\}$. So we can continue as before, provided that we show $\mathcal{U}_n$ and $\mathcal{W}_{\sigma(n)}$ for all $n \in \omega$.

Suppose $\mathcal{U}_n \cong \mathcal{L}_m$. For each $k \in \omega$ define the formula

$$\delta_k(\overline{x}) =_{\text{def}} \exists! w \left(\rho_{\overline{a}|_n}(w, \overline{x}) \wedge \varphi_{\mathcal{L}_k}(w)\right) \tag{6.48}$$

where $\varphi_{\mathcal{L}_k}$ is as in Lemma 6.2.6. Then we have that, for all $k \in \omega$, $\mathcal{T} \models \delta_k(\overline{a}|_n)$ if and only if $k = m$, and hence $\mathcal{S} \models \delta_k(\overline{b}|_n)$ if and only if $k = m$. Therefore, $\mathcal{W}_{\sigma(n)} \cong \mathcal{L}_m$.

Case $\mathcal{U} \cong \mathcal{A}_{\omega+1}$: The proof for this case is analogous to that of the case $\mathcal{U} \cong \mathcal{E}_{\omega+1}$.

Case $\mathcal{U} \cong \mathcal{A}_{\omega+2}$: All of the limbs $\mathcal{U}_n$ are isomorphic to $\mathcal{E}_{\omega+1}$. Hence the proof is analogous to that of the case $\mathcal{U} \cong \mathcal{A}_{k+1}$. $\square$

The case for $\mathcal{U} \cong \mathcal{A}_{\omega+2}$ is exactly what we need to prove the homogeneity of $\mathcal{A}_{\omega+2}$. Hence Theorem 6.4.1 follows immediately from the following proposition.

Proposition 6.4.13. *$\mathcal{A}_{\omega+2}$ is homogeneous.*

Proof. Suppose $(\mathcal{A}_{\omega+2}, \overline{a}) \equiv (\mathcal{A}_{\omega+2}, \overline{b})$. By setting $\mathcal{T} = \mathcal{S} = \mathcal{A}_{\omega+2}$ in Lemma 6.4.12, there is an automorphism of $\mathcal{A}_{\omega+2}$ taking $\overline{a}$ to $\overline{b}$. So $\mathcal{A}_{\omega+2}$ is strongly homogeneous, and hence homogeneous. $\square$

6.5 Atomic structures

For any structure $\mathfrak{A}$ and $\overline{a}, \overline{b} \in \mathfrak{A}$,

$$\text{type}(\overline{a}) = \text{type}(\overline{b}) \quad \Leftrightarrow \quad (\mathfrak{A}, \overline{a}) \equiv (\mathfrak{A}, \overline{b}) \tag{6.49}$$

Therefore, we should be able to us the back-and-forth trees analyze the complexity of properties whose definition includes some mention of types. The simplest such is the property that of being atomic.

Definition 6.5.1. A structure $\mathfrak{A}$ is *atomic* if for all $\overline{a} \in \mathfrak{A}$, type($\overline{a}$) is a principal type in Th($\mathfrak{A}$).

Given this definition, we can define the predicate

$$\text{ATOM}(e) =_{\text{def}} \mathfrak{A}_e \text{ is an atomic structure} \tag{6.50}$$

From the definition, it is clear that

$$\begin{aligned}\text{ATOM}(e) \Leftrightarrow \forall \overline{a} \in \mathfrak{A}_e\, \exists(\varphi \in \mathcal{L}) \Big(&\big(\mathfrak{A}_e \vDash \varphi(\overline{a})\big) \wedge \\ &\forall(\psi \in \mathcal{L}) \Big[\big(\mathfrak{A}_e \vDash \psi(\overline{a})\big) \to \big(\mathfrak{A}_e \vDash \forall \overline{x}\, (\varphi(\overline{x}) \to \psi(\overline{x}))\big)\Big]\Big)\end{aligned} \tag{6.51}$$

The second conjunct of (6.51) can be rewritten as

$$\bigwedge_{n \in \omega} \forall \psi(\overline{x}) \in \exists_n \Big[\big(\mathfrak{A}_e \vDash \psi(\overline{a})\big) \to \big(\mathfrak{A}_e \vDash \forall \overline{x}\, (\varphi(\overline{x}) \to \psi(\overline{x}))\big)\Big] \tag{6.52}$$

It is clear then from (6.51) and (6.52) that ATOM is at most $\Pi^0_{\omega+2}$. Hence our goal of this section is to prove the following.

Theorem 6.5.2. *The class of computable atomic structures is $\Pi^0_{\omega+2}$-complete.*

Unlike the previous section, we cannot directly use the structures $\mathcal{E}_{\omega+2}$ and $\mathcal{A}_{\omega+2}$ to prove Theorem 6.5.2; neither of these trees is atomic. The following proposition illustrates why this is.

Proposition 6.5.3. *Let $\mathcal{T}$ be any tree with an $\mathcal{E}_{\omega+1}$ limb. Then $\mathcal{T}$ is not atomic.*

Proof. Let $a \in \mathcal{T}$ be the root of the $\mathcal{E}_{\omega+1}$ limb, and let b be the root of an $\mathcal{L}_\infty$ limb attached to a. We will show that type(b) is not principal.

Let $\varphi(x) \in$ type(b) be any formula, and let m be the quantifier rank of φ. Let c be the root of an $\mathcal{L}_m$ limb attached to a. Using Lemma 6.4.7 to define our strategy in the Ehrenfeucht-Fraïssé game, it is easy to see that $(\mathcal{T}, b) \equiv_m (\mathcal{T}, c)$. So $\varphi(x) \in$ type(c).

However, type(c) $\neq$ type(b) by Lemma 6.2.6. So $\varphi(x)$ cannot be the principal generator of type(b). □

On the other hand, it turns out that $\mathcal{E}_{\omega+1}$ is the least of the back-and-forth trees that is not atomic. Clearly a formula generates the type of some $a \in \mathfrak{A}$ if it defines those elements up to automorphism in $\mathfrak{A}$. Modifying the proof of Lemma 6.4.12, we can get the following result.

Lemma 6.5.4. *Let $\mathcal{T}$ be a tree, $\mathcal{S}$ an $\mathcal{L}_k$ limb with root $a \in \mathcal{T}$, for $k \in \omega \cup \{\infty\}$. Then, for all $\overline{b} \in \mathcal{S}$, there is a formula $\varrho_{\overline{b}}(x, \overline{y})$ such that for all $\overline{c} \in \mathcal{T}$,*

$$\mathcal{T} \vDash \varrho_{\overline{b}}(a, \overline{c}) \quad \Leftrightarrow \quad (\mathcal{T}, \overline{b}) \cong (\mathcal{T}, \overline{c}) \tag{6.53}$$

Proof. Let $\mathcal{S}_n$ be an enumeration of the limbs of $\mathcal{S}$. As $\overline{b}$ is a finite sequence, the $\overline{b}|_n = \emptyset$ for all but finitely many $n \in \omega$. Therefore, we need only show that, whenever $\overline{b}|_n \neq \emptyset$, there is a formula $\varrho_{\overline{b}|_n}$ satisfying (6.53). Then, by Lemma 6.4.11, the conjunction of these formulae will give us our desired formula $\varrho_{\overline{b}}$.

Take any $n \in \omega$ and assume that $\mathcal{S} \cong \mathcal{E}_k$; the proof for $\mathcal{S} \cong \mathcal{A}_k$ is analogous. We define the formula

$$\varrho_{\overline{b}|_n}(x, \overline{y}) =_{\text{def}} \rho_{\overline{b}|_n}(x, \overline{y}) \wedge \varphi_{\mathcal{E}_k}(x) \wedge \psi_k^{\text{len}}(x) \tag{6.54}$$

where $\rho_{\overline{b}|_n}$ is as in (6.41) and ψ_k^{len} is as in (6.13) of Lemma 6.2.6. As $\mathcal{L}_k$ has exactly one limb isomorphic $\mathcal{E}_l$ attached to a, it is clear from the proof of Lemma 6.4.12 that this formula identifies $\overline{b}|_n$ up to automorphism in $\mathcal{T}$. ☐

Corollary 6.5.5. *$\mathcal{A}_{\omega+1}$ is atomic.*

Proof. Let be any sequence of constants in an $\mathcal{L}_k$ limb attached to the root of $\mathcal{A}_{\omega+1}$, for $k \in \omega$. By Lemma 6.4.11, it is enough show that there is a formula $\varphi_{\overline{a}}$ which defines $\overline{a}$ up to automorphism in $\mathcal{A}_{\omega+1}$.

By Lemma 6.2.6, it is clear that the formula

$$\psi(x) =_{\text{def}} \exists y \left(\varphi^{\text{root}}(y) \wedge y \; R \; x \wedge \varphi_{\mathcal{L}_k}(x)\right) \tag{6.55}$$

identifies the root of this $\mathcal{L}_k$ limb up to automorphism in $\mathcal{A}_\omega$. So, by Lemma 6.5.4, the formula

$$\varphi_{\overline{a}}(\overline{x}) =_{\text{def}} \exists w \left(\rho_{\overline{a}}(w, \overline{x}) \wedge \psi(w)\right) \tag{6.56}$$

identifies $\overline{b}$ up to automorphism in $\mathcal{A}_{\omega+1}$. ☐

Unfortunately, Corollary 6.5.5 is not enough to give us Theorem 6.5.2; it only shows that ATOM is $\Pi^0_{\omega+1}$-hard. To see how we can push this result up another quantifier level, we need to understand why $\mathcal{E}_{\omega+1}$ fails to be atomic. Because of Lemma 6.5.4, we can identify all of the elements inside the $\mathcal{L}_k$ limbs of $\mathcal{E}_{\omega+1}$ up to isomorphism in $\mathcal{L}_k$. Our problem is that we don't have a single formula that can distinguish $\mathcal{L}_\infty$ from the other limbs in $\mathcal{E}_{\omega+1}$.

This problem is not simply a result of the fact that $\mathcal{E}_{\omega+1}$ has an $\mathcal{L}_\infty$ limb attached to the root, while $\mathcal{E}_{\omega+1}$ does not. By Lemma 6.5.4, the tree with only $\mathcal{L}_\infty$ limbs attached to the root is still atomic. $\mathcal{E}_{\omega+1}$ is not atomic because, in addition, it has $\mathcal{L}_m$ limbs attached to the root for arbitrarily large $m \in \omega$. If we were to create a tree that looked like $\mathcal{E}_{\omega+1}$, but which only had finitely many $\mathcal{L}_m$ limbs for $m \in \omega$, we would not have this problem. This is the technique we will use to prove Theorem 6.5.2.

For the remainder of this section, we let $\mathcal{P}(n)$ be a fixed $\Pi^0_{\omega+2}$ predicate. We can write this predicate as

$$\mathcal{P}(n) = \forall x \exists y \; \mathcal{Q}(x, y, n) = \forall x \exists y \bigwedge_{i \in \omega} \mathcal{Q}_i(x, y, n) \tag{6.57}$$

where $\mathcal{Q}$ is Π^0_ω and the $\mathcal{Q}_i$ are each Π^0_i. Using these predicates, we recursively define a new sequence of predicates

(6.58)
$$\mathcal{S}_i(x,y,n) \quad \Leftrightarrow \quad \begin{cases} \textbf{True} & \text{if } y=-1 \text{ or } i=-1 \\ \mathcal{S}_{i-1}(x,y-1,n) \vee \bigwedge_{j=1}^{i} \mathcal{Q}_j(x,y,n) & \text{otherwise} \end{cases}$$

These predicates "uniformize" the $\mathcal{Q}_i$ in some sense, as shown in the following lemma.

Lemma 6.5.6. *For any $x, y, n \in \omega$, there is some $m \in \omega \cup \{\infty\}$ such that*

$$\mathcal{S}_i(x,y,n) \quad \Leftrightarrow \quad i < m \tag{6.59}$$

Moreover, $m = \infty$ if and only if $\mathcal{Q}(x,z,n)$ for some $z \leq y$.

Proof. Take any x, $n \in \omega$. We proceed by induction on y.

Case $y = 0$: By the definition of $\mathcal{S}_i$,

$$\mathcal{S}_i(x,0,n) \quad \Leftrightarrow \quad \bigwedge_{j=1}^{i} \mathcal{Q}_j(x,0,n) \tag{6.60}$$

for all $i \in \omega$. If $\neg\mathcal{Q}(x,0,n)$, there is some least $m \in \omega$ such that $\neg\mathcal{Q}_m(x,0,n)$; it is clear that this m satisfies (6.59). Otherwise, if $\mathcal{Q}(x,0,n)$, then $\mathcal{S}_i(x,0,n)$ holds for all $i \in \omega$, and so $m = \infty$ works.

Case $y = k+1$: By the induction hypothesis, there is some m_0 witnessing (6.59) for k. First suppose that $\neg\mathcal{Q}(x,z,n)$ for all $z \leq y$. Then $m_0 \neq \infty$, and there is some least $m_1 \in \omega$ such that $\neg\mathcal{Q}_{m_1}(x,y,n)$. Set $m = \max(m_0+1, m_1)$. It is clear from (6.58) that this choice of m works.

Otherwise, suppose that $\mathcal{Q}(x,z,n)$ for some $z \leq y$. Then $m_0 = \infty$, and so $\mathcal{S}_i(x,k,n)$ holds for all i. Then it is clear from (6.58) that $\mathcal{S}_i(x,y,n)$ holds for all $i \in \omega$. ☐

We denote the unique m in Lemma 6.5.6 by $m^{x,n}_y$. The following properties of $m^{x,n}_y$ are clear from the proof of Lemma 6.5.6.

Lemma 6.5.7. *Take any x and $n \in \omega$, and define the function $f(y) = m^{x,n}_y$. Then f has the following properties.*

(a) $f(y) \leq f(y+1)$ for all $y \in \omega$.

(b) $f(y) = f(y+1)$ only if $f(y) = \infty$.

(c) $f(y) = \infty$ for all but finitely many $y \in \omega$ if and only if $\exists y\, Q(x,y,n)$.

Now we define some more trees to code this new predicate.

Definition 6.5.8. Take any $k, n \in \omega$. $\mathcal{B}_n^k$ is the tree with the following properties.

(a) $\mathcal{B}_n^k$ has exactly one node of depth m for each $m \leq k$. The node of depth k is called the *branching node of* $\mathcal{B}_n^k$.

(b) $\mathcal{B}_n^k$ has infinitely many $\mathcal{L}_{m_y^{k,n}}$ limbs attached to the branching node, for each $y \in \omega$

(c) $\mathcal{B}_n^k$ has infinitely many $\mathcal{L}_\infty$ limbs attached branching node.

$\mathcal{B}_n$ is the tree with exactly one copy of $\mathcal{B}_n^x$ limb attached to the root for each $x \in \omega$. These trees are illustrated in Figure 6.6.

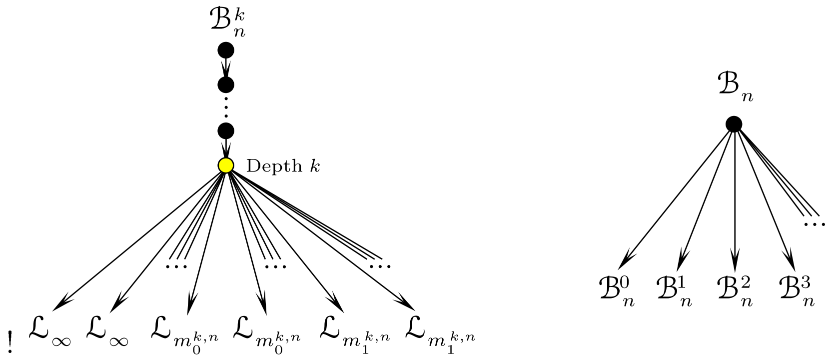

Figure 6.6: The trees $\mathcal{B}_n^k$ and $\mathcal{B}_n$

The branching nodes are used to distinguish the limbs $\mathcal{B}_n^k$ from each other in $\mathcal{B}_n$. So, provided that we can define the elements in the limbs of $\mathcal{B}_n$ up to automorphism, $\mathcal{B}_n$ is atomic.

Proposition 6.5.9. *Suppose $\mathcal{P}(n)$ holds. Then for all $\overline{a} \in \mathcal{B}_n$, there is a formula $\varrho_{\overline{a}}(x)$ such that for all $\overline{b} \in \mathcal{B}_n$,*

$$\mathcal{B}_n \vDash \varrho_{\overline{a}}(\overline{b}) \quad \Leftrightarrow \quad (\mathcal{B}_n, \overline{a}) \cong (\mathcal{B}_n, \overline{b}) \tag{6.61}$$

Proof. The proof of this claim is similar to that of Lemma 6.5.4. Take any $\overline{a} \in \mathcal{B}_n$. As the root of $\mathcal{B}_n$ is definable by the formula φ^{root}, and the sequence $\overline{a}$ is finite, it is enough to consider the elements $\overline{a}|_k$ in one of the limbs $\mathcal{B}_n^k$. By Lemma 6.4.11, the conjunction of φ^{root} and the formulae $\varrho_{\overline{a}|_k}$ will give us our desired formula $\varrho_{\overline{a}}$.

Take any $k \in \omega$, and let $\overline{c} = \overline{a}|_k$. Also let b be the branching node of $\mathcal{B}_n^k$ in $\mathcal{B}_n$. We define the formula

(6.62)
$$\delta_b(x) =_{\text{def}} \exists y \left(\varphi^{\text{root}}(y) \wedge (y\ R^k\ x) \wedge \forall u, v \bigwedge_{i<k} \left((y\ R^i\ u) \wedge (y\ R^i\ v) \to u = v\right)\right) \wedge$$
$$\exists z, w \left((x\ R\ z) \wedge (x\ R\ w) \wedge z \neq w\right)$$

Each limb $\mathcal{B}_n^\ell$ attached the root of $\mathcal{B}_n$ has its branching node at depth $\ell + 1$ in $\mathcal{B}_n$. So clearly δ_b identifies b up to automorphism in $\mathcal{B}_n$. As there are only $k-1$ elements above b in $\mathcal{B}_n$, this means that we can define any elements of $\overline{c}$ above b up to automorphism.

So again, by Lemma 6.4.11, it is enough for us to consider the case when the $\overline{c}$ are all contained in a single limb attached to b. Suppose $\overline{c} \in \mathcal{L}_\ell$ for some $\ell \in \omega \cup \{\infty\}$. As $\mathcal{P}(n)$ holds, so does $\exists y\, \mathcal{Q}(k, y, n)$. We have two cases.

Case $\ell \neq \infty$: We define the formula

$$\varrho_{\overline{c}}(\overline{x}) =_{\text{def}} \exists y \left(\delta_b(y) \wedge \rho_{\overline{c}}(x, \overline{y}) \wedge \varphi_{\mathcal{L}_\ell}(x)\right) \tag{6.63}$$

where $\rho_{\overline{c}}$ is as in (6.41) of Lemma 6.4.11. As in the proof of Lemma 6.5.4, it is clear that this choice of $\varrho_{\overline{c}}$ defines $\overline{c}$ up to automorphism in $\mathcal{B}_n$.

Case $\ell = \infty$: By Lemma 6.5.7, there are only finitely many $m_0 < \ldots < m_s \in \omega$ such that the branching node $\mathcal{B}_n^k$ has an $\mathcal{L}_{m_t}$ limb. We define the formula

$$\varrho_{\overline{c}}(\overline{x}) =_{\text{def}} \exists y \left(\delta_b(y) \wedge \rho_{\overline{c}}(x, \overline{y}) \wedge \bigwedge_{t \leq s} \neg\varphi_{\mathcal{L}_{m_t}}(x)\right) \tag{6.64}$$

where $\rho_{\overline{c}}$ is as in (6.41). For any $\overline{d} \in \mathcal{B}_n$ such that $\mathcal{B}_n \models \varrho_{\overline{c}}(\overline{d})$, the elements $\overline{d}$ cannot be contained in an $\mathcal{L}_{m_t}$ limb for $t \leq s$. Hence it is clear that this choice of $\varrho_{\overline{c}}$ defines $\overline{c}$ up to automorphism in $\mathcal{B}_n$. □

Corollary 6.5.10. *If $\mathcal{P}(n)$, then $\mathcal{B}_n$ is atomic.*

Proof. For any $\overline{a} \in \mathcal{B}_n$, the formula $\varrho_{\overline{a}}$ is the principal generator of type($\overline{a}$). □

So, to prove Theorem 6.5.2, we need only show the converse.

Proposition 6.5.11. *$\mathcal{B}_n$ is not atomic for all n such that $\neg\mathcal{P}(n)$.*

Proof. This proof is similar to Proposition 6.5.3. Fix $n \in \omega$ such that $\neg\mathcal{P}(n)$, and let k be least such that $\forall y\, \neg\mathcal{Q}(k, y, n)$. Let a be the branching node of $\mathcal{B}_n^k$ in $\mathcal{B}_n$, and let b be the root of an $\mathcal{L}_\infty$ limb attached to a. We will show that type(b) is not principal.

Let $\varphi(x) \in$ type(b) be any formula, and let m be the quantifier rank of φ. By Lemma 6.5.7, there is some y such that $m_y^{k,n} \geq m+1$. Let c be the root of an $\mathcal{L}_{m_y^{k,n}}$ limb attached to a. As in Proposition 6.5.3, it is easy to see that $(\mathcal{B}_n, b) \approx_m (\mathcal{B}_n, c)$. So $\varphi(x) \in$ type(c). However, type(c) $\neq$ type(b) by Lemma 6.2.6. So $\varphi(x)$ cannot be the principal generator of type(b). □

6.6 Locally matched structures

The Ehrenfeucht-Fraïssé games in section 6.4 were fairly straight-forward. We were able to define our strategies on the back-and-forth trees by composing strategies on the limbs. However, as we saw in section 6.5, we sometimes have to modify the back-and-forth trees in order to analyze the complexity of a model theoretic property. Therefore, we may have to develop more complicated strategies for these structures.

In this section we will develop a collection of strategies that will be helpful in analyzing computable saturation in section 6.7. Our strategies will take advantage of the fact that our structures are all graphs with respect to R. As the games all have finite length, player$_\forall$ cannot force a relation between two elements if they are too far away from each other. This is why the back-and-forth trees agree on sentences of increasing quantifier rank as they get larger.

Therefore, it seems that we should be able to define a strategy based upon the local nature of our graphs. This is the motivation for the following definition.

Definition 6.6.1. Let $\mathfrak{A}$ be a structure in our language $\mathcal{L}$ of a single binary relation R. For $a, b \in \mathfrak{A}$, we define the relation $a \sim_n b$ by induction on n.

- $a \sim_0 b$ if and only if $a = b$.
- $a \sim_{n+1} b$ if and only if there is some $c \in \mathfrak{A}$ such that $a \sim_n c$ and either $(c\ R\ b)$, $(b\ R\ c)$, or $b = c$.

For $a \in \mathfrak{A}$, $n \in \omega$, the *n-ball of a* is the set

$$O(a, n) = \{\, b \mid a \sim_n b \,\} \tag{6.65}$$

Suppose that we are playing the game $\mathrm{EF}_n(\mathfrak{A}, \mathfrak{B})$ and player$_\forall$ initially picks some element $a \in \mathfrak{A}$. Also suppose that we can choose some $k \in \omega$ and $b \in \mathfrak{B}$ such that $O(a, k) \cong O(b, k)$. Then we can win any game in which player$_\forall$ restricts himself to choosing from $O(a, k)$ and $O(b, k)$. If we can generalize this idea to arbitrary collections of k-balls, then we should be able convert this into a winning strategy for $\mathrm{EF}_n(\mathfrak{A}, \mathfrak{B})$. This is the motivation for the following definition.

Definition 6.6.2. A *local matching* for the structures $\mathfrak{A}$, $\mathfrak{B}$ is a triple $\sigma = (\overline{p}, \overline{q}, \overline{r})$ with the following properties.

(a) The sequences $\overline{p}$, $\overline{q}$, and $\overline{r}$ all have the same length.

(b) $\overline{p} \in \mathfrak{A}$, $\overline{q} \in \mathfrak{B}$, and $\overline{r} \in \omega$.

(c) For all $i \leq |\overline{r}|$, $O(p_i, r_i) \cong O(q_i, r_i)$.

(d) For all $i \neq j$, $O(p_i, r_i) \cap O(p_j, r_j)$, $O(q_i, r_i) \cap O(q_j, r_j) = \emptyset$.

For any pair $(\overline{a}, \overline{b})$, with $\overline{a} \in \mathfrak{A}$ and $\overline{b} \in \mathfrak{B}$ and $|\overline{a}| = |\overline{b}|$, we say that the local matching σ *safely covers* $(\overline{a}, \overline{b})$, if the following additional properties hold.

(e) For all $i \leq |\overline{a}|$, there is a $j \leq |\overline{r}|$ such that $a_i \in O(p_j, r_j - 1)$

(f) For all $i \leq |\overline{a}|$ and all $j \leq |\overline{r}|$, $a_i \in O(p_j, r_j - 1)$ if and only if $b_i \in O(q_j, r_j - 1)$.

(g) For each $j \le |\overline{r}|$, there is an isomorphism $f : O(p_j, r_j) \to O(q_j, r_j)$ such that $f(a_i) = b_i$ for all $a_i \in O(p_j, r_j)$.

Lemma 6.6.3. *Let $(\overline{c}, \overline{d})$ be a play of the game* $\mathrm{EF}_n((\mathfrak{A}, \overline{a}), (\mathfrak{B}, \overline{b}))$*, and $(\overline{p}, \overline{q}, \overline{r})$ be a local matching pair that safely covers $(\overline{ac}, \overline{bd})$. Then the play $(\overline{c}, \overline{d})$ is a win for player$_\exists$.*

Proof. We can assume without loss of generality that $\overline{c} \supseteq \overline{a}$, $\overline{d} \supseteq \overline{b}$. Take any c_i, $c_j \in \overline{c}$ such that $\mathfrak{A} \models (c_i\ R\ c_j)$. By properties (e) and (d), there is some $k \le |\overline{r}|$ such that c_i, $c_j \in O(p_k, r_k)$. So d_i, $d_j \in O(q_k, r_k)$ by property (f). Hence $\mathfrak{B} \models (d_i\ R\ d_j)$ by property (g).

By a similar argument in the other direction, we have that

$$\mathfrak{A} \models (c_i\ R\ c_j) \Leftrightarrow \mathfrak{B} \models (d_i\ R\ d_j) \tag{6.66}$$

for all c_i, $c_j \in \overline{c}$. So $(\overline{c}, \overline{d})$ is a win for player$_\exists$. □

In order to get a local matching that safely covers our play, we need to be able to extend our local matching whenever player$_\forall$ chooses an element that is outside the matching. This is the motivation for the following definition.

Definition 6.6.4. For any $c \in \mathfrak{A}$, and $n \in \omega$, we say that the local matching σ *partially n-extends to c* if $O(c, n) \cap O(p_i, n_i) = \emptyset$ for all $i \le |\overline{r}|$. We define what it means for a local matching to partially n-extend to $d \in \mathfrak{B}$ in the analogous way.

For $\overline{a} \in \mathfrak{A}$ and $\overline{b} \in \mathfrak{B}$, we say that $(\mathfrak{A}, \overline{a})$ and $(\mathfrak{B}, \overline{b})$ are *locally matched up to* $n \in \omega$, which we denote $(\mathfrak{A}, \overline{a}) \stackrel{\circ}{=}_n (\mathfrak{B}, \overline{b})$, if the following all hold.

(a) For any $n \in \omega$, there is a matching pair $(\overline{p}, \overline{q}, \overline{r})$ which safely covers $(\overline{a}, \overline{b})$ such that $r_i = 2^{3n+1}$ for all $i \le |\overline{r}|$.

(b) Let $\overline{a} \subseteq \overline{c} \in \mathfrak{A}$, $\overline{b} \subseteq \overline{d} \in \mathfrak{B}$ with $\sigma = (\overline{p}, \overline{q}, \overline{r})$ a local matching which safely covers $(\overline{c}, \overline{d})$. For any $c_0 \in \mathfrak{A}$ and $1 \le k \le 2^{3n+1}$ such that σ partially k-extends to c_0, there is some $d_0 \in \mathfrak{B}$ and local matching $\tau = (\overline{p}p_0, \overline{q}q_0, \overline{r}k)$ which safely covers $(\overline{c}c_0, \overline{d}d_0)$.

(c) Let $\overline{a} \subseteq \overline{c} \in \mathfrak{A}$, $\overline{b} \subseteq \overline{d} \in \mathfrak{B}$ with $\sigma = (\overline{p}, \overline{q}, \overline{r})$ a local matching which safely covers $(\overline{c}, \overline{d})$. For any $d_0 \in \mathfrak{B}$ such that σ partially n-extends to d_0, $n \ge 1$, there is some $c_0 \in \mathfrak{A}$ and local matching $\tau = (\overline{p}p_0, \overline{q}q_0, \overline{r}n)$ which safely covers $(\overline{c}c_0, \overline{d}d_0)$.

If $\mathfrak{A} \stackrel{\circ}{=}_n (\mathfrak{B}, \overline{b})$ for all $n \in \omega$, we write $\mathfrak{A} \stackrel{\circ}{=} (\mathfrak{B}, \overline{b})$

While the definition of $\stackrel{\circ}{=}$ may seem fairly complicated, the idea behind it is rather simple. If $\mathfrak{A} \stackrel{\circ}{=} \mathfrak{B}$, then we can play the Ehrenfeucht-Fraïssé game $\mathrm{EF}_n(\mathfrak{A}, \mathfrak{B})$ by constructing local matchings which safely cover our play at each stage of the game. So long as player$_\forall$ chooses elements inside the local matching, we play the strategy induced by property (g) of Definition 6.6.2. If player$_\forall$ chooses an element outside of our local matching, we can use properties (b) and (c) of Definition 6.6.4 to extend our local matching. By formalizing this strategy, we get the following result.

Theorem 6.6.5. *Let $\overline{a} \in \mathfrak{A}$, $\overline{b} \in \mathfrak{B}$ be such that $(\mathfrak{A}, \overline{a}) \mathrel{\mathring{=}}_n (\mathfrak{B}, \overline{b})$. Then $(\mathfrak{A}, \overline{a}) \approx_n (\mathfrak{B}, \overline{b})$.*

Proof. For simplicity, we now define a play at stage $k \leq n$ to be both the elements chosen by player$_\forall$ and player$_\exists$ up to that stage, and the elements $\overline{a} \in \mathfrak{A}$, $\overline{b} \in \mathfrak{B}$. That way, when we refer to a play $(\overline{c}, \overline{d})$ of the game at stage k, we can assume that $\overline{a} \subseteq \overline{c}$, $\overline{b} \subseteq \overline{d}$.

As we said above, at each stage $k \leq n$, we want to construct a local matching $\sigma_k = (\overline{p}, \overline{q}, \overline{r})$ which safely covers the play of the game at that stage. Our primary obstacle is when player$_\forall$ plays at the borders of our n-balls in this local matching. If $(\overline{c}, \overline{d})$ is the play at stage k, property (e) of Definition 6.6.2 requires that for all $i \leq |\overline{c}|$ there is some $j \leq |\overline{r}|$ such that $c_i \in O(p_i, r_j - 1)$. Suppose that, at the next stage, player$_\forall$ chooses some $e \in \mathfrak{A}$ with $p_0 \sim_{r_j} e$. Then σ_k is not partially n-extendable to e for any $n \in \omega$, and we are not guaranteed a choice for $f \in \mathfrak{B}$ such that we can safely cover $(\overline{c}e, \overline{d}f)$. This problem is illustrated in Figure 6.7

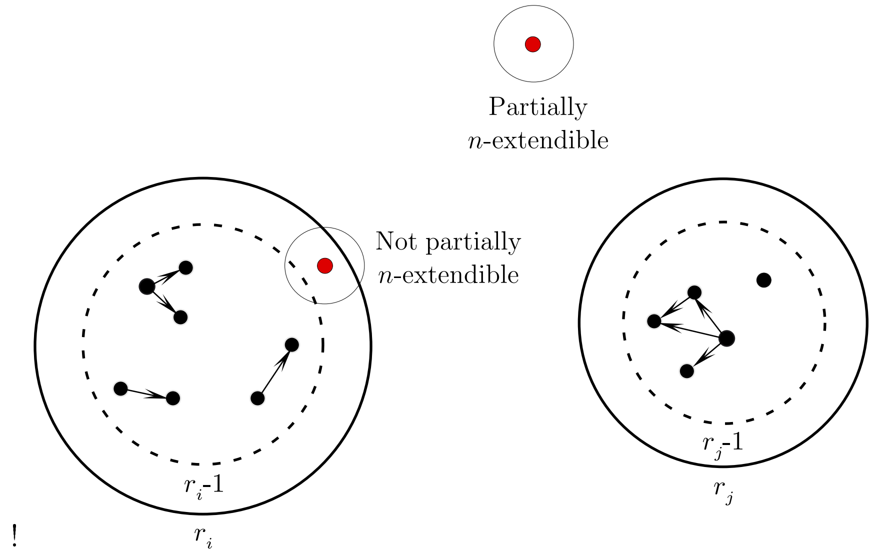

Figure 6.7: The primary obstacle to local matching

To solve this problem, we will require our local matching σ_k to have the following properties, in addition to safely covering the play at stage k.

(a) For all $j \leq |\overline{r}|$, $2^{3(n-k)+1} \leq r_j$.

(b) Let $(\overline{c}, \overline{d})$ is the play of the game by stage $k \leq n$. then for all $i \leq \overline{c}$, there is some $j \leq |\overline{r}|$ such that

$$c_i \in O(p_j, r_j - 2^{3(n-k)}) \quad \text{and} \quad d_i \in O(q_j, r_j - 2^{3(n-k)}) \tag{6.67}$$

Property (b) will allow us to dynamically change the borders of the n-balls in our local matchings to prevent the type of problem illustrated in Figure 6.7. With this in mind, we define our strategy as follows.

Stage $k = -1$:
Before the start of the game, we choose a local matching $\sigma_{-1} = (\overline{p}, \overline{q}, \overline{r})$ which safely covers $(\overline{a}, \overline{b})$ and satisfies the two properties (a) and (a) above. We can do this by the definition of $(\mathfrak{A}, \overline{a}) \stackrel{\circ}{=} (\mathfrak{B}, \overline{b})$. By property (a) of Definition 6.6.4, we can choose the radii $\overline{r} \in \omega$ large enough such that for all $a_i \in \overline{a}$, there is some $k \leq |\overline{r}|$ with

$$a_i \sim_n a_j \Leftrightarrow a_j \in O(p_k, r_k) \quad \text{for all } n \in \omega,\ a_j \in \overline{a} \tag{6.68}$$

Enlarging these radii $\overline{r}$ even further guarantees properties (a) and (b) above.

Stage $k = \ell + 1$:
Let $(\overline{c}, \overline{d})$ be the play by stage ℓ, and let $\sigma_\ell = (\overline{p}, \overline{q}, \overline{r})$. At this stage, player$_\forall$ picks either some $c_k \in \mathfrak{A}$ or some $d_k \in \mathfrak{B}$. We can assume that player$_\forall$ picks $c_k \in \mathfrak{A}$; the proof for when player$_\forall$ picks $d_k \in \mathfrak{A}$ is analogous. We have two cases.

Case $c_k \in O(p_i, r_i - 2^{3(n-\ell)-1})$ for some $i \leq |\overline{r}|$: We let $f : O(p_i, r_i) \to O(q_i, r_i)$ be the isomorphism from property (g) of Definition 6.6.2. We choose $b_k = f(a_k)$ for our next move in the game, and set $\sigma_k = \sigma_\ell$.

The isomorphism $f : O(c_i, r_i) \to O(d_i, r_i)$ preserves the relations $\sim_s$ on $O(p_i, r_i)$ for $s \leq r_i$. So, as $r_i - 2^{3(n-\ell)-1} = r_i - 2^{3(n-k)+2}$,

$$b_k \in O(p_i, r_i - 2^{3(n-\ell)-1})) \subseteq O(p_i, r_i - 2^{3(n-k)}) \tag{6.69}$$

Hence σ_k is a local matching safely covering $(\overline{c}c_k, \overline{d}d_k)$ with properties (a) and (b) above.

Case $a_k \notin O(p_j, r_j - 2^{3(n-\ell)-1})$ for all $j \leq |\overline{r}|$: Let $m = |\overline{r}|$. We define the sequence $\overline{q} \in \omega$ by

$$q_j = \begin{cases} r_j - (2^{3(n-\ell)-1} + 2^{3(n-k)+1}) & \text{If } j \leq m \\ 2^{3(n-k)+1} & \text{If } j = m+1 \end{cases} \tag{6.70}$$

These elements are all well-defined by property (a) above Let $\tau = (\overline{p}, \overline{q}, \overline{q})$. For all $j \leq m$,

$$\begin{aligned} q_j = r_j - (2^{3(n-\ell)-1} + 2^{3(n-k)+1}) &= r_j - (2^{3(n-k)+2} + 2^{3(n-k)+1}) \\ > r_j - 2^{3(n-k)+3} = 2^{3(n-k)+3} &= r_j - 2^{3(n-\ell)} \end{aligned} \tag{6.71}$$

So τ safely covers $(\overline{c}, \overline{d})$ by property (b) above.

Suppose that there is some element $e \in O(a_k, q_{m+1}) \cap O(c_j, q_j)$ then $a_k \in O(c_j, q_{m+1} + q_j)$. Then, $q_{m+1} + q_j = r_j - 2^{3(n-\ell)-1}$ by (6.70), which is a contradiction. Hence τ partial q_{m+1} extends to a_k. We choose b_k and let

$$\sigma_k = (\overline{p}p_k, \overline{q}q_k, \overline{q}) \tag{6.72}$$

be a local matching safely covering $(\overline{c}c_k, \overline{d}d_k)$. This move in the game is illustrated in Figure 6.8

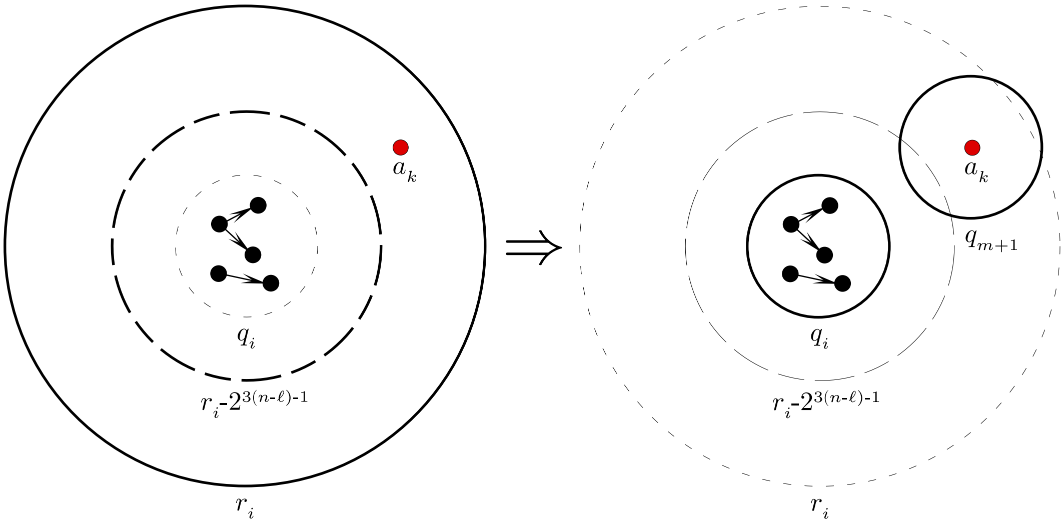

Figure 6.8: The move for $a_k \notin O(p_j, r_j - 2^{3(n-\ell)-1})$

We need only show that σ_k has the properties (a) and (b) above. For all $j \leq m$,

$$\begin{aligned} q_j &= r_j - (2^{3(n-\ell)-1} + 2^{3(n-k)+1}) = r_j - (2^{3(n-k)+2} + 2^{3(n-k)+1}) \\ &\geq 2^{3(n-\ell)+1} - 2^{3(n-k)+3} = 2^{3(n-k)+3} \end{aligned} \tag{6.73}$$

So property (a) holds. Furthermore, for all $j \leq m$.

$$q_j - 2^{3(n-k)} = r_j - (2^{3(n-\ell)-1} + 2^{3(n-\ell)-2}) \leq r_j - 2^{3(n-\ell)} \tag{6.74}$$

So property (b) holds by induction.

Given this strategy, we let $(\overline{c}, \overline{d})$ be any play defined by this strategy with σ_n the local matching safely covering this pair. By Lemma 6.6.3, this play is a win for player$_\exists$. $\square$

6.7 Computable saturation

As we said in section 6.5, we should be able to use the back-and-forth trees to analyze properties defined in terms of the types of the structure. We have already used them to analyze the complexity of being an atomic structure. However, we cannot use them to analyze all properties defined in terms of types. For example, we cannot use them to analyze the complexity of saturation. Proposition 6.2.3 only gives us reductions for hyperarithmetic predicates, and we have already seen in Chapter 5 that saturation is Π^1_1-complete.

However, there is a effective variant of saturation.

Definition 6.7.1. Let $\mathfrak{A}$ be a structure in a computable language $\mathcal{L}$. $\mathfrak{A}$ is *computably saturated* if for all $\overline{a} \in \mathfrak{A}$ and computable sets of formula $\Gamma(x, \overline{a})$, there is some $b \in \mathfrak{A}$ realizing Γ.

Given this definition, we define the following predicate.

$$\text{SAT}_c(e) =_{\text{def}} \mathfrak{M}_e \text{ is computably saturated} \tag{6.75}$$

For each sequence of variables $\overline{y}$, let $\phi_n(x, \overline{y})$ be an enumeration of the formulae with free variables x and $\overline{y}$. Then it is clear that

(6.76)
$$\text{SAT}_c(e) \Leftrightarrow \forall i, \overline{a} \left(\forall n \left(\mathfrak{M}_e \models \exists x \bigwedge_{\substack{k \leq n \\ \{i\}(k)=1}} \phi_k(x, \overline{a}) \right) \to \exists b \left(\mathfrak{M}_e \models \bigwedge_{\{i\}(n)=1} \phi_n(b, \overline{a}) \right) \right)$$

So, as with ATOM, it is clear that SAT_c is at most $\Pi^0_{\omega+2}$. Our goal for this section is to show that this calculation is a lower bound as well.

Again, we would like to use our back-and-forth trees $\mathcal{E}_{\omega+2}$ and $\mathcal{A}_{\omega+1}$. Note that model $\mathcal{E}_{\omega+2}$ is not computably saturated. Let $a \in \mathcal{E}_{\omega+2}$ be a child of the root whose limb is a copy of $\mathcal{A}_{\omega+1}$. Consider the computable set of formulae

$$\Gamma(x) = \{ (a \; R \; x) \} \cup \{ \neg\varphi_{\mathcal{L}_m}(x) \}_{m \in \omega} \tag{6.77}$$

By compactness and Lemma 6.2.6, this set of formulae is consistent with $\mathcal{E}_{\omega+2}$. However, it is not realized in $\mathcal{E}_{\omega+2}$, since the realizing element cannot be the root of an $\mathcal{L}_m$ tree for $m \in \omega$.

Therefore, we would be done if $\mathcal{A}_{\omega+2}$ were computably saturated. Unfortunately, this is false, because of the following result.

Proposition 6.7.2. *Let Ω be the full ω-branching tree; that is, it is the tree in which every element has countably many children. For any countable tree $\mathcal{S}$ and $m \in \omega \cup \{ \infty \}$, $\mathcal{S} \equiv \mathcal{L}_m$ if and only if $\mathcal{S}$ is a copy of $\mathcal{L}_m$ with at most countably many copies of Ω attached to the root.*

Proof. ($\Leftarrow$) Let $\mathcal{S}$ be a copy of $\mathcal{L}_m$ with at most countably copies of Ω attached to the root. Let a be the root of $\mathcal{S}$ and b the root of $\mathcal{L}_m$. By Theorem 6.6.5, we need only show that $(\mathcal{S}, a) \stackrel{\circ}{=} (\mathcal{L}_m, b)$.

Let $n \in \omega$. For each $c \in \mathcal{S}$, we define

$$\mathcal{S}^c_n = \{ x \mid (c \; R^n \; x) \} \tag{6.78}$$

$\mathcal{S}^c_n$ is the restriction of the limb with root c to depth n. By assumption, for all but finitely many children c of a, $\mathcal{S}^c_n$ is the full ω-branching tree of depth n; that is, every element in this tree with depth less than n has infinitely many children, and every element with depth n has no children. The only c for which $\mathcal{S}^c_n$ is not the full ω-branching tree of depth n are those that are the roots of $\mathcal{A}_k$, $\mathcal{E}_k$ limbs for $k < n + 1$.

A similar result holds for the children of b in $\mathcal{L}_m$. So, as $\mathcal{S}$ is just $\mathcal{L}_m$ with copies of Ω attached to the root, we have that $O(a, n) \cong O(b, n)$. Furthermore, as

a and b each have no parent, any such isomorphism must necessarily take a to b. Hence, as $n \in \omega$ was arbitrary, we can satisfy property (a) of Definition 6.6.4.

Next, we need to show that we can satisfy property (b) of Definition 6.6.4. Let $\overline{c} \in \mathcal{S}$, $\overline{d} \in \mathcal{L}_m$ with $a \in \overline{c}$ and $b \in \overline{d}$, and let $\sigma = (\overline{p}, \overline{q}, \overline{r})$ be a local matching safely covering $(\overline{c}, \overline{d})$. Suppose σ partially n-extends to $c_0 \in \mathcal{S}$. It is clear from Definition 6.6.1 that there is a unique element $c_1 \in O(c_0, n)$ with no parent; c_1 is just the ancestor of c_0 such that $(c_1\ R^n\ c_0)$. We let $\mathcal{U}$ be the limb with root c_1. We have three cases.

Case $\mathcal{U} \cong \mathcal{E}_k$: It is clear from the definition of $\mathcal{L}_m$ that we can choose some $\mathcal{E}_k$ limb of $\mathcal{L}_m$ which does not intersect any of the sets $O(q_i, r_i)$. Let d_0 be the element corresponding to c_0 in the isomorphism from $\mathcal{U}$ to this limb. Then $O(d_0, n) \cap O(q_i, r_i) = \emptyset$ for all $i \leq |\overline{r}|$, and so $\tau = (\overline{p}c_0, \overline{q}d_0, \overline{r}n)$ satisfies property (b).

Case $\mathcal{U} \cong \mathcal{A}_k$: This proof of this case is analogous to that of $\mathcal{U} \cong \mathcal{E}_k$.

Case $\mathcal{U} \cong \Omega$: We have already seen that for each $k \in \omega$, we can choose some $\mathcal{E}_k$ limb of $\mathcal{L}_m$ which does not intersect any of the sets $O(q_i, r_i)$. Let d_0^k be the element corresponding to c_0 in the isomorphism from $\mathcal{U}$ to each such limb. If we choose some $k \geq 2(n+1)$, then there are no childless descendants of d_n^k less than n steps away from d_0^k. Therefore, there is an isomorphism $f : O(c_0, n) \to O(d_0^k, n)$ which takes c_0 to d_0^k. Hence the local matching $\tau = (\overline{p}c_0, \overline{q}d_0^k, \overline{r}n)$ satisfies property (b).

The proof of property(c) in Definition 6.6.4 is analogous. So $(\mathcal{S}, a) \mathrel{\dot{=}} (\mathcal{L}_m, b)$.

($\Rightarrow$) First we introduce some new formulae. We define the formula

$$\chi_k^{\mathrm{dscd}}(x) =_{\mathrm{def}} \exists y \forall z \big((x\ R^k\ y) \wedge (\neg x\ R^{k-1}\ y) \wedge (\neg y\ R\ z)\big) \tag{6.79}$$

χ_k^{dscd} expresses that there is at least one childless element exactly k steps away from x. It is clear from the definition of the back-and-forth trees that for any $c \in \mathcal{L}_m$ and limb $\mathcal{U}$ with root c,

$$\mathcal{L}_m \models \chi_k^{\mathrm{dscd}}(c) \quad \Rightarrow \quad ES \ncong \mathcal{A}_\ell, \mathcal{E}_\ell \text{ for } \ell \geq k+2 \tag{6.80}$$

Now we define the formula

$$\chi_{\leq m}^{\mathrm{bnd}}(x) =_{\mathrm{def}} \forall y, z \big((x\ R^m\ y) \rightarrow (\neg y\ R\ z)\big) \tag{6.81}$$

$\chi_{\leq m}^{\mathrm{bnd}}$ expresses that there are no elements $m+1$ steps away from x. It follows from (6.80) and the definition of the back-and-forth structures that for any $c \in \mathcal{L}_m$

$$\mathcal{L}_m \models \chi_k^{\mathrm{dscd}}(c) \quad \Rightarrow \quad \mathcal{L}_m \models \chi_{\leq k+1}^{\mathrm{bnd}}(c) \tag{6.82}$$

By construction, for each $\ell \in \omega$, there are only finitely many elements c attached to the root of $\mathcal{L}_m$ such that $\mathcal{L}_m \models \chi_{\leq \ell}^{\mathrm{bnd}}(c)$.

Let b be the root if $\mathcal{L}_m$, and let d be any child of b. We know from the proof of Lemma 6.2.6 that there is a unique formula $\varphi_d(x)$ that defines the limb with root d up to isomorphism. As $\mathcal{S} \equiv \mathcal{L}_m$, there is some child $c \in \mathcal{S}$ of a, with $\mathcal{S} \models \varphi_d(c)$.

Therefore the limb with root c is isomorphic to the limb with root d. As all of the limbs attached to the root of $\mathcal{L}_m$ are distinct, this implies that $\mathcal{L}_m$ is embeddable in $\mathcal{S}$

Now let c be any child of a that is not part of the embedding of $\mathcal{L}_m$ into $\mathcal{S}$. Take an $\ell \in \omega$. By (6.82), there are only finitely many children $d \in \mathcal{L}_m$ of b such that $\mathcal{L}_m \models \chi_\ell^{\mathrm{dscd}}(d)$. So, as $\mathcal{S} \equiv \mathcal{L}_m$, $\mathcal{S} \not\models \chi_\ell^{\mathrm{dscd}}(b)$. As $\ell \in \omega$, it is clear from (6.79) that the limb with root c is a copy of Ω. $\square$

Because of this, we will again have to modify our back-and-forth trees to calculate the complexity of SAT_c. We will modify our trees by constructing a computable structure $\mathfrak{M}^+$, elementarily equivalent to $\mathcal{A}_{\omega+2}$, that is computably saturated. In order to do this, we will first need better understand the computable types of $\mathrm{Th}(\mathcal{A}_{\omega+2})$.

The analysis would be much simpler if we needed to consider only the computable complete types of $\mathrm{Th}(\mathcal{A}_{\omega+2})$. However, computable saturation is defined for all computable sets of formulae consistent with $\mathrm{Th}(\mathfrak{A}, \overline{a})$, not just complete types. So it is not immediately obvious that a computable set of formulae $\Gamma(x, \overline{a})$ consistent with $\mathrm{Th}(\mathfrak{A}, \overline{a})$ is contained in even a hyperarithmetic complete type.

However, when $\mathrm{Th}(\mathfrak{A}, \overline{a})$ is decidable, we can uniformly extend Γ to a Δ_2^0 complete type of $\mathrm{Th}(\mathfrak{A}, \overline{a})$. We set $\Phi_0 = \Gamma$ and enumerate the formulae $\phi_n(x)$ of $\mathcal{L}(\overline{a})$ with free variable x. At each stage s, since the theory is decidable and Γ_s is computable, we can use $0'$ to compute whether or not $\Phi_s \cup \{ \phi_n \}$ is consistent with $\mathrm{Th}(\mathfrak{A}, \overline{a})$; we extend Φ_s to Φ_{s+1} accordingly.

With this fact in mind, our first goal of this section will be to prove the following result.

Theorem 6.7.3. $\mathrm{Th}(\mathcal{A}_{\omega+2})$ *is decidable.*

We will prove this theorem by giving a computable list of axioms for $\mathrm{Th}(\mathcal{A}_{\omega+2})$. In order to axiomatize $\mathrm{Th}(\mathcal{A}_{\omega+2})$, we first introduce some new sets of formulae. We define the formula

$$\chi_{=m}^{\mathrm{bnd}}(x) =_{\mathrm{def}} (\neg\chi_{\leq m-1}^{\mathrm{bnd}}(x) \wedge \chi_{\leq m}^{\mathrm{bnd}}(x)) \tag{6.83}$$

We also let $\varphi_{\mathcal{E}_k}$ be as in Lemma 6.2.5 and $\varphi_{\mathcal{L}_k}$ be as in Lemma 6.2.6. Then we define the following for all $k \in \omega$.

$$A_1(x) = \{ \forall y\, (\neg x \ R \ y) \} \tag{6.84}$$

$$E_1(x) = \{ \exists^{\geq n} y\, (x \ R \ y) \}_{n\in\omega} \cup \{ \forall y, z\, (x \ R \ y \to \neg y \ R \ z) \} \tag{6.85}$$

$$A_{k+1}(x) = \{ \exists^{\geq n} y\, (x \ R \ y) \}_{n\in\omega} \cup \{ \forall y\, (x \ R \ y \to \psi(y)) \mid \psi(y) \in E_k(y) \} \tag{6.86}$$

$$\begin{aligned} E_{k+1}(x) = {} & \{ \exists^{\geq n} y\, (x \ R \ y \wedge \varphi_{\mathcal{E}_k}(y)), \exists^{\geq n} y\, (x \ R \ y \wedge \neg\varphi_{\mathcal{E}_k}(y)) \}_{n\in\omega} \cup \\ & \{ \forall y\, ((x \ R \ y \wedge \varphi_{\mathcal{E}_k}(y)) \to \psi(y)) \mid \psi(y) \in E_k(y) \} \cup \\ & \{ \forall y\, ((x \ R \ y \wedge \neg\varphi_{\mathcal{E}_k}(y)) \to \psi(y)) \mid \psi(y) \in A_k(y) \} \end{aligned} \tag{6.87}$$

$$(6.88)\qquad \begin{aligned} L_k(x) = &\{\, \exists! y\,\big((x\ R\ y) \wedge \chi^{\mathrm{bnd}}_{=m}(y)\big)\,\}_{m\in\omega} \cup \\ &\{\, \forall y\,\big((x\ R\ y \wedge \chi^{\mathrm{dscd}}_{k}(x)) \to \chi^{\mathrm{bnd}}_{\leq m+1}(x)\big)\,\}_{m\in\omega} \cup \\ &\{\, \forall y\,\big((x\ R^m\ y) \wedge \exists^{\geq n} z(y\ R\ z)\big) \to \exists^{\geq n+1} w(y\ R\ w)\,\}_{n,m\in\omega} \cup \\ &\{\, \forall y\,\big(((x\ R\ y) \wedge \chi^{\mathrm{bnd}}_{=m}(x)) \to \psi(y)\big) \mid \psi(y) \in A_m(y), m \leq k\,\} \\ &\{\, \forall y\,\big(((x\ R\ y) \wedge \chi^{\mathrm{bnd}}_{=k}(x)) \to \psi(y)\big) \mid \psi(y) \in E_m(y), m > k\,\} \end{aligned}$$

These sets of formulae have the following properties.

Proposition 6.7.4. *Let $\mathcal{T}$ be a countable tree. Then the following hold for any $a \in \mathcal{T}$.*

(a) $\mathcal{T} \vDash A_k(a)$ if and only if the limb with root a is isomorphic to $\mathcal{A}_k$.

(b) $\mathcal{T} \vDash E_k(a)$ if and only if the limb with root a is isomorphic to $\mathcal{E}_k$.

(c) $\mathcal{T} \vDash L_m(a)$ if and only if the limb with root a is elementarily equivalent to $\mathcal{L}_m$.

Proof. It is clear from the definitions in (6.84) and (6.85) that the claims (a) and (b) both hold for $k = 1$. Therefore, claims (a) and (b) hold for all $k \in \omega$ by induction on the definition of the back-and-forth trees.

Note that it is also clear from (6.88) that $\mathcal{T} \vDash L_m(a)$ if and only if the limb with root a is a copy of $\mathcal{L}_m$ with at most countably copies of Ω attached to a. Therefore, claim (c) follows from Proposition 6.7.2. □

Given these properties, we can now list our axioms for $\mathrm{Th}(\mathcal{A}_{\omega+2})$. The theory $T^{\omega+2}$ is given by the following axioms.

(a) *There are no cycles.* We express this by the collection of sentences $\forall x\,(\neg x\ R^n\ x)$ with $n \geq 1$.

(b) *Each element has at most one predecessor.* We express this by the sentence

$$(6.89)\qquad \forall x, y, z\,\big(((y\ R\ x) \wedge (z\ R\ x)) \to y = x\big)$$

Given this axiom and the requirement that there are no cycles, we can define the parent and children of any element in the usual way.

(c) *Each element has either no children, or infinitely many children.* We express this by the collection of sentences

$$(6.90)\qquad \forall x\,\big(\exists^{\geq n} y(x\ R\ y) \to \exists^{\geq n+1} y(x\ R\ y)\big)$$

(d) *There is only one element with no parent, which we refer to as the root; moreover, this root has infinitely many children.* We express this by the collection of sentences

$$(6.91)\qquad \exists! x\, \varphi^{\mathrm{root}}(x) \wedge \exists x\, \exists^{\geq n} y\,(\varphi^{\mathrm{root}}(x) \wedge (x\ R\ y))$$

with $n \in \omega$. Given the uniqueness of this root, we can define the depth of any element to be its distance from the root. This depth will be infinite if the element is not in the same connected component as the root.

(e) *Any element that does not have depth* 2 *has branches of uniformly bounded length.* We express this by the collection of sentences

$$\forall x \left(\neg\chi^{\text{depth}}_{\leq 2}(x) \to (\chi^{\text{dscd}}_{n}(x) \to \chi^{\text{bnd}}_{\leq n+1}(x))\right) \tag{6.92}$$

and

$$\forall x \left((\neg\chi^{\text{depth}}_{\leq 2}(x) \wedge (x\ R\ y) \wedge (x\ R\ z)) \to (\chi^{\text{bnd}}_{\leq n}(y) \leftrightarrow \chi^{\text{bnd}}_{\leq n}(z))\right) \tag{6.93}$$

for all $n \geq 2$, with χ^{depth}_{d} as in (6.39). Note that we do not include $n = 1$ because $\mathcal{A}_1$ has no children, while $\mathcal{E}_1$ has infinitely many children.

(f) *For any element with depth greater than* 2, *that element is the root of either a copy of* Ω, $\mathcal{E}_n$, *or* $\mathcal{A}_n$ *for some* $n \in \omega$. Together (6.92) and (6.93) guarantee that, if c does not satisfy $\chi^{\text{bnd}}_{=n}(x)$ for any $n \in \omega$, then it must be the root of a copy of Ω. Therefore, we can express this requirement by the collection of sentences

$$\forall x \left((\chi^{\text{bnd}}_{=n}(x) \wedge \varphi_{\mathcal{E}_n}(x)) \to \psi(x)\right) \tag{6.94}$$

for $\psi \in E_n(x)$ with $n \geq 1$, and

$$\forall x \left((\chi^{\text{bnd}}_{=n}(x) \wedge \neg\varphi_{\mathcal{E}_k}(x)) \to \psi(x)\right) \tag{6.95}$$

for $\psi \in A_k(x)$ with $n \geq 2$. Note that $(\chi^{\text{bnd}}_{=1}(x) \wedge \neg\varphi_{\mathcal{E}_k}(x))$ is inconsistent; $\mathcal{A}_1$ limbs are trivially definable, as they consist of a single element.

(g) *If a element of depth* 2 *satisfies* $\varphi_{\mathcal{L}_n}$, *then the limb with that root is elementarily equivalent to* $\mathcal{L}_m$. By Proposition 6.7.4, we can express this by the collection of sentences

$$\forall x \left((\chi^{\text{depth}}_{2}(x) \wedge \varphi_{\mathcal{L}_m}(x)) \to \psi(x)\right) \tag{6.96}$$

for all $\psi \in L_n(x)$

(h) *Any element of depth* 2 *which does not satisfy* $\varphi_{\mathcal{L}_n}$ *for any* $n \in \omega$ *is elementarily equivalent to* $\mathcal{L}_\infty$. By Lemma 6.4.7, we know that $\mathcal{L}_\infty$ agrees with $\mathcal{L}_n$ for on sentences of quantifier rank up to n. So by Proposition 6.7.4, we can express this by the collection of sentences

$$\forall x \left(\left(\chi^{\text{depth}}_{2}(x) \wedge \neg \bigvee_{k \leq m} \varphi_{\mathcal{L}_k}(x)\right) \to \psi(x)\right) \tag{6.97}$$

for all $\psi \in \mathcal{L}_n(x)$ with quantifier rank less than n.

(i) *The children of the root have infinitely many limbs elementarily equivalent to $\mathcal{L}_m$ for each $m \in \omega$.* Given the axioms in (6.96), we can express this by the collection of sentences

$$\forall x \left(\chi_1^{\text{depth}}(x) \to \exists^{\geq n} y((x\ R\ y) \wedge \varphi_{\mathcal{L}_m}(y))\right) \tag{6.98}$$

with n, $m \in \omega$.

It is clear from the discussion in our axiomatization that $\mathcal{A}_{\omega+2} \models T^{\omega+2}$, and so this is a consistent theory. The list of axioms is computable, so this is in fact a c.e. theory. If we can show that $T^{\omega+2}$ is complete, the $\mathrm{Th}(\mathcal{A}_{\omega+2}) = T^{\omega+2}$ and hence $T^{\omega+2}$ is decidable. So Theorem 6.7.3 is a corollary of the theorem below.

Definition 6.7.5. Let $\mathfrak{B} \models T^{\omega+2}$. The *primary tree* of $\mathfrak{B}$ is the connected component containing the root. We denote this substructure as $\mathcal{T}_{\mathfrak{B}}$. Any connected component of $\mathfrak{B}$ other than the primary tree is a *nonstandard component.*

Theorem 6.7.6. *Any two models of $T^{\omega+2}$ are elementarily equivalent.*

Proof. Let $\mathfrak{B} \models T^{\omega+2}$. It is enough to show that $\mathfrak{B} \equiv \mathcal{A}_{\omega+2}$. Let a be the root node of $\mathcal{A}_{\omega+2}$ and b the root node of $\mathfrak{B}$. Again, we need only show that $(\mathcal{A}_{\omega+2}, a) \triangleq (\mathfrak{B}, b)$.

First we need to show we can satisfy property (a) of Definition 6.6.4. Let d be any child b, with $\mathcal{T}_{\mathfrak{B}}^d$ the limb with root d in $\mathcal{T}_{\mathfrak{B}}$. By the axioms in (6.96) and (6.97), any child of d is the root of some limb elementarily equivalent to $\mathcal{L}_m$ for some $m \in \omega \cup \{\infty\}$. Furthermore, the axioms in (6.98) guarantee that, for each $m \in \omega$, there are infinitely many children of d that are the root of a limb elementarily equivalent to $\mathcal{L}_m$. So, by modifying the strategy in Lemma 6.4.8, we see that $\mathcal{T}_{\mathfrak{B}}^d \equiv \mathcal{E}_{\omega+1}$.

Given this fact, it is easy to see that $\mathcal{T}_{\mathfrak{B}} \equiv \mathcal{A}_{\omega+2}$. We adapt our strategy for the game $\mathrm{EF}_n(\mathcal{T}, \mathcal{A}_{\omega+2})$ from Proposition 6.4.9; we just mate each $\mathcal{T}_{\mathfrak{B}}^d$ to some copy of $\mathcal{E}_{\omega+1}$ in $\mathcal{A}_{\omega+2}$, and conversely, as dictated by the moves of player$_\forall$. Since $\mathcal{T}_{\mathfrak{B}}^d \equiv \mathcal{E}_{\omega+1}$ for arbitrary children d of b, this is a winning strategy.

As $\mathcal{T}_{\mathfrak{B}} \equiv \mathcal{A}_{\omega+2}$, it is clear that $O(a,n) \cong O(b,n)$ for all $n \in \omega$; both $O(a,n)$ and $O(b,n)$ are finite graphs and hence definable up to isomorphism by a first order sentence. Furthermore, any such isomorphism must necessarily take a to b. Hence we can satisfy property (a) of Definition 6.6.4.

Next we need to satisfy property (b) of Definition 6.6.4. Let $\overline{c} \in \mathcal{A}_{\omega+2}$, $\overline{d} \in \mathfrak{B}$ with $a \in \overline{c}$ and $b \in \overline{d}$, and let $\sigma = (\overline{p}, \overline{q}, \overline{r})$ be a local matching safely covering $(\overline{c}, \overline{d})$. Suppose σ partially n-extends to $c_0 \in \mathcal{A}_{\omega+2}$. Let c_1 be the unique element of $O(c_0, n)$ with no predecessor.

Since σ safely covers $(\overline{c}, \overline{d})$ and $a \in \overline{c}$, there is some $i \leq |\overline{r}|$ such that $O(p_i, r_i)$ contains a and all of its children. So as $O(p_i, r_i) \cap O(c_0, n) = \emptyset$, c_1 has depth at least 2. If c_1 has depth 2, then the limb with root c_1 is elementarily equivalent to $\mathcal{L}_m$ for some $m \in \cup \{\infty\}$. Then, as in the proof of Proposition 6.7.2, we can satisfy property (b) by choosing some $\mathcal{L}_m$ limb of $\mathcal{A}_{\omega+2}$ which does not intersect any of the $O(q_i, r_i)$.

If c_1 has depth greater than 2, then the axioms in (6.94) and (6.95) guarantee that it is the root of a copy of Ω, $\mathcal{E}_k$, or $\mathcal{A}_k$ for some $k \in \omega$. In this case we satisfy property (b) exactly as we did in Proposition 6.7.2.

The proof that we can satisfy (c) of Definition 6.6.4 is analogous. So $(\mathcal{A}_{\omega+2}, a) \mathrel{\mathring{=}} (\mathfrak{B}, b)$, and hence $\mathfrak{B} \equiv \mathcal{A}_{\omega+2}$. □

Unfortunately, even though $T^{\omega+2}$ is decidable, this does not necessarily mean that for any $\mathfrak{B} \equiv \mathcal{A}_{\omega+2}$ and $\overline{b} \in \mathfrak{B}$, $\mathrm{Th}(\mathfrak{B}, \overline{b})$ is decidable. The elements $\overline{b}$ could be elements in a fairly complicated nonstandard component, and hence type$(\overline{b})$ may not be computable. Therefore, our next step is to categorize the nonstandard components of models of $T^{\omega+2}$.

Definition 6.7.7. We construct the acyclic graph Ω^* as follows.

(a) The set $\{ a_n \}_{n \in \omega} \in \Omega^*$ is an ascending chain in this graph. That is, $\Omega^* \models (a_{n+1} \; R \; a_n)$ for all $n \in \omega$.

(b) Every element of Ω^* has a_n as an ancestor for some $n \in \omega$.

(c) For all $n \in \omega$, the limb with root a_n is a copy of Ω.

This graph is illustrated in Figure 6.9.

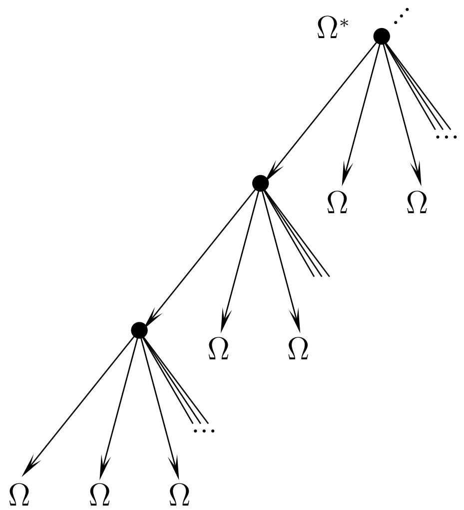

Figure 6.9: The graph Ω^*

Ω^* is a fairly simply structure; there is only one automorphism orbit of this structure. Therefore, nonstandard components that are isomorphic to Ω^* will not be a problem. We will be more concerned with the following structures.

Definition 6.7.8. Given a function $fin2^\omega$, we say that f is *consistent with* $T^{\omega+2}$ if $f(n+1) = 1$ whenever $f(n) = 0$. For any f consistent with $T^{\omega+2}$, we construct the acyclic graph $\mathcal{E}^f$ as follows.

(a) The set $\{ a_n \}_{n\in\omega} \in \mathcal{E}^f$ is an ascending chain in this graph. That is, $\mathcal{E}^f \models (a_{n+1} \; R \; a_n)$ for all $n \in \omega$.

(b) Every element of $\mathcal{E}^f$ has a_n as an ancestor for some $n \in \omega$.

(c) If $f(n) = 0$, the limb with root a_n is a copy of $\mathcal{A}_{n+1}$. Otherwise it is a copy of $\mathcal{E}_{n+1}$.

This graph is illustrated in Figure 6.10.

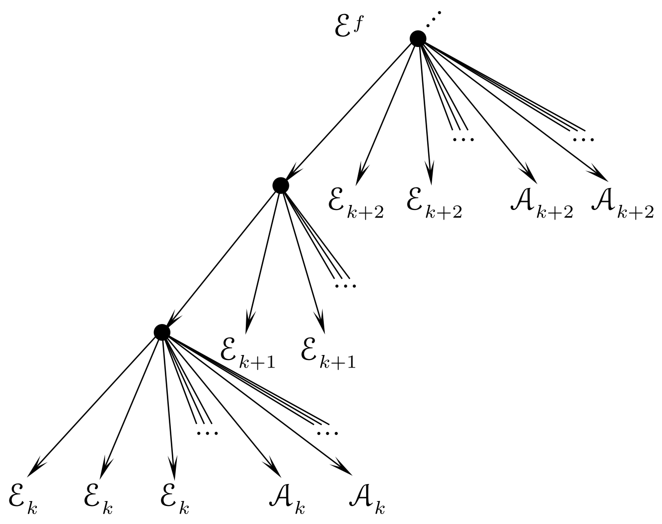

Figure 6.10: The graph $\mathcal{E}^f$

We specified that our functions f be consistent with T^ω simply for convenience; by definition, there are no $\mathcal{A}_n$ limbs attached to the root of $\mathcal{A}_{n+1}$. However, it is clear that we can transform any $f \in 2^\omega$ into a function g consistent with $T^{\omega+2}$ such that $f = g$ whenever f is also consistent. So from now on, we will refer to the graph $\mathcal{E}^f$ for any function f.

As the following result shows, these graphs are precisely the nonstandard components of models of $T^{\omega+2}$.

Proposition 6.7.9. *Let $\mathfrak{B} \models T^{\omega+2}$. Any nonstandard component of $\mathfrak{B}$ is isomorphic to Ω^* or $\mathcal{E}^f$, for some $f \in 2^\omega$.*

Proof. Let $a \in \mathfrak{B}$ be an element in a nonstandard component, and let $\mathcal{S}$ be the limb with root a. Since a is not on the primary tree, a has infinite depth. So by the axioms in (6.94) and (6.95) we have three cases for $\mathcal{S}$.

Case $\mathcal{S} \cong \Omega$: Since a is not on the primary tree, every ancestor of a also has infinite depth. The axioms in (6.92) and (6.93) guarantee that $\mathfrak{B}$ branches uniformly below each of these ancestors, and so these ancestors must all be the root of a full ω-branching tree. So by definition, a is in a copy of Ω^*.

Case $\mathcal{S} \cong \mathcal{E}_k$ **for some** k: Again, the axioms in (6.92) and (6.93) guarantee that $\mathfrak{B}$ branches uniformly below each of the ancestors of a. In particular, for each ancestor of a, there is some $n \in \omega$ such that this ancestor satisfies $\chi^{\text{bnd}}_{=n}(x)$. So by the axioms in (6.94) and (6.95), each ancestor is either a copy of $\mathcal{E}_\ell$ or $\mathcal{A}_\ell$ for some $\ell \in \omega$. So by definition, a is in a copy of $\mathcal{E}^f$.

Case $\mathcal{S} \cong \mathcal{A}_k$ **for some** k: The proof for this case is analogous to the case $\mathcal{T} \cong \mathcal{E}_k$. □

Given this proposition, we can now define our candidate for a computably saturated model.

Definition 6.7.10. For each $m \in \omega \cup \{\infty\}$, $\mathcal{L}^+_m$ is a copy of $\mathcal{L}_m$ with countably many copies of Ω attached to the root. Similarly, $\mathcal{E}^+_{\omega+1}$ is the tree with infinitely many $\mathcal{L}^+_m$ limbs attached to the root for each $m \in \omega \cup \{\infty\}$, and $\mathcal{A}^+_{\omega+2}$ is the tree with infinitely many $\mathcal{E}^+_{\omega+1}$ limbs attached to the root. These structures are illustrated in Figure 6.11.

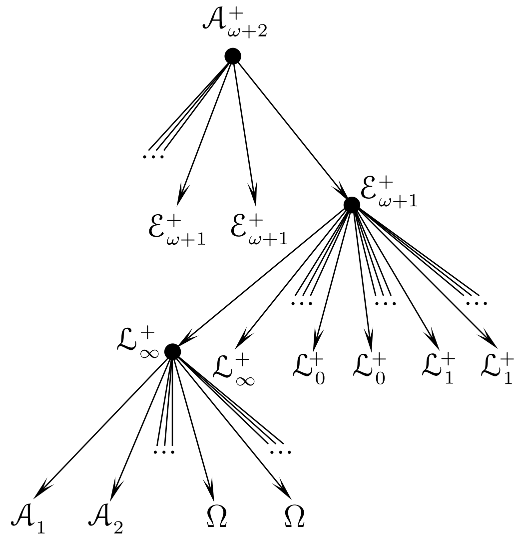

Figure 6.11: The structure $\mathcal{A}^+_{\omega+2}$

The structure $\mathfrak{M}^+$ is the disjoint union of $\mathcal{A}^+_{\omega+2}$, infinitely many copies of $\mathcal{E}^f$ for each arithmetic function $f \in 2^\omega$, and infinitely many copies of Ω^*.

Since we are adding structures $\mathcal{E}^f$ for all arithmetic f, it is not immediately obvious that $\mathfrak{M}^+$ is even computably presentable. Fortunately, the following result gives us a way to computably present $\mathfrak{M}^+$.

Lemma 6.7.11. *Let f, $g \in 2^\omega$ be functions such that f and g agree on all but finitely many values. Then $\mathcal{E}^f \cong \mathcal{E}^g$.*

Proof. Let $\{ a_n \}_{n\in\omega}$ be the ascending sequence defining $\mathcal{E}^f$, $\{ b_n \}_{n\in\omega}$ the sequence defining $\mathcal{E}^g$. Let k be greatest element such that $f(k) \neq g(k)$. We can assume without loss of generality that $f(k) = 0$. Then $f(k+1) = g(k+1) = 1$.

So the limb with root b_{k+1} has an $\mathcal{A}_k$ limb attached to the root. Hence for $n \leq k$, there are elements c_n in this copy of $\mathcal{A}_k$ such that the limb with root c_n is isomorphic to the limb with root a_n. We define the mapping

$$a_n \mapsto \begin{cases} b_n & n > k \\ c_n & n \leq k \end{cases} \tag{6.99}$$

It is clear that this induces an isomorphism. $\square$

Proposition 6.7.12. *$\mathfrak{M}^+$ is computably presentable.*

Proof. The structures $\mathcal{A}^+_{\omega+2}$ and Ω^* are clearly computable. We need only show that, for any arithmetic f, we can construct $\mathcal{E}^f$ uniformly in an index for f. Take any f and suppose that f is Σ^0_n for some $n \in \omega$. Then, by Proposition 6.2.3, we can uniformly construct the limb with root a_k for each $k > n$. By Lemma 6.7.11, this is enough. $\square$

From now on we will identify $\mathfrak{M}^+$ with its computable presentation. It is clear that $\mathfrak{M}^+ \models T^{\omega+2}$. So given the decidability of $T^{\omega+2}$, we can now characterize the types of $\mathfrak{M}^+$.

Theorem 6.7.13. *For any $\overline{a} \in \mathfrak{M}^+$,* $\mathrm{type}(\overline{a})$ *is arithmetic.*

Proof. For simplicity, we will first prove this claim for the case when $\overline{a}$ is a single element a. Then we will show how to extend it to an arbitrary finite sequence of elements $\overline{a}$.

We will construct an arithmetic set of sentences $\Gamma_a(x)$ such that $\Gamma_a(x)$ uniquely determines $\mathrm{type}(a)$. Since $\mathrm{Th}(\mathfrak{M}^+) = T^{\omega+2}$ is decidable, this will imply that $\mathrm{type}(a)$ is arithmetic. Let $\mathcal{S}$ be the limb with root a. Our definition will depend on $\mathcal{S}$ and the depth of a in $\mathfrak{M}^+$.

Case a has depth $k \leq 1$: All of the children of the root are in the same automorphism orbit. So in this case, $\Gamma_a = \left\{ \chi^{\mathrm{depth}}_k(x) \right\}$ is sufficient.

Case a has depth 2: Then $\mathcal{S} \cong \mathcal{L}^+_m$ for some $m \in \omega \cup \{ \infty \}$. It is clear from the axioms in (6.96) that, if $m \in \omega$, the set

$$\Gamma_a(x) = \left\{ \chi^{\mathrm{depth}}_2(x) \right\} \cup \{ \varphi_{\mathcal{L}_m}(x) \} \tag{6.100}$$

is sufficient. Otherwise, we let

$$\Gamma_a(x) = \left\{ \chi^{\mathrm{depth}}_k(x) \right\} \cup \{ \neg\varphi_{\mathcal{L}_m}(x) \}_{m\in\omega} \tag{6.101}$$

Case a has finite depth $k > 2$, $\mathcal{S} \cong \Omega$: Let b be the ancestor of a with depth 2. Then the limb with root b is isomorphic to $\mathcal{L}_m^+$ for some $m \in \omega \cup \{\infty\}$. As $\mathcal{S} \cong \Omega$, a must be in one of the Ω limbs of $\mathcal{L}_m^+$. In this case it is clear that a is defined up to automorphism by type(b) and its relationship to b. We define

$$\Gamma_a(x) = \left\{ \exists x \, (\chi_2^{\text{depth}}(y) \wedge (y \; R^{k-2}) \wedge \psi) \;\middle|\; \psi \in \Gamma_b(y) \right\} \tag{6.102}$$

This is a computable set of formulae since $\Gamma_b(y)$ is.

Case a has finite depth $k > 2$, $\mathcal{S} \cong \mathcal{E}_\ell$: This case is similar to the case when $\mathcal{S} \cong \Omega$. Let $b_0, \ldots, b_{k-1}$ be the ancestors of a. It is clear from the proof of Lemma 6.4.12 that a is defined up to automorphism by the formula $\varphi_{\mathcal{E}_\ell}$, the types type$(b_i)$, and their relationship to a. We define

$$\Gamma_a(x) = \{ \varphi_{\mathcal{E}_\ell}(x) \} \cup \left\{ \exists x \, (\chi_2^{\text{depth}}(y) \wedge (y \; R^{k-i}) \wedge \psi) \;\middle|\; \psi \in \Gamma_{b_i}(y), i < k \right\} \tag{6.103}$$

By induction on the depth of the b_i, we see that Γ_{b_i} is computable for all $i < k$. Hence Γ_a is computable.

Case a has finite depth $k > 2$, $\mathcal{S} \cong \mathcal{A}_\ell$: In this case, we replace $\varphi_{\mathcal{E}_\ell}$ in (6.103) with $\neg\varphi_{\mathcal{E}_\ell}$.

Case a is in a copy of Ω^*: All elements of Ω^* are automorphic to any other element of Ω^*. Therefore, it is clear that a is defined up to isomorphism by the set

$$\Gamma_a(x) = \left\{ \neg\chi_k^{\text{depth}}(x) \right\}_{k \in \omega} \cup \{ \neg\chi_k^{\text{dscd}}(x) \}_{k \in \omega} \tag{6.104}$$

Case a is in a copy of $\mathcal{E}^f$: Assume that $\mathcal{S} \cong \mathcal{E}_\ell$; the proof for $\mathcal{S} \cong \mathcal{A}_\ell$ is analogous. Let $\{ b_n \}_{n \in \omega}$ be an enumeration of a and its ancestors, such that $b_0 = a$ and $(b_n \; R^n \; a)$ for all $n > 0$. Let $\mathcal{S}_n$ be the limb with root b_n for each $n \in \omega$. It is clear from Definition 6.7.8 that $\mathcal{E}^f$ is determined up to isomorphism by the limbs $\mathcal{S}_n$.

Axioms (6.94) and (6.95) guarantee that there are formulae ϕ_n such that for all $c \in \mathfrak{M}$, $\mathfrak{M} \vDash \phi_n(c)$ if and only if the limb with root c is isomorphic to $\mathcal{S}_n$. In particular

$$\phi_n(x) =_{\text{def}} \begin{cases} (\chi_{=(n+\ell)}^{\text{bnd}}(x) \wedge \varphi_{\mathcal{E}_{(n+\ell)}}(x)) & \text{if } \mathcal{S}_n \cong \mathcal{E}_{n+\ell} \\ (\chi_{=(n+\ell)}^{\text{bnd}}(x) \wedge \varphi_{\mathcal{E}_{(n+\ell)}}(x)) & \text{otherwise} \end{cases} \tag{6.105}$$

Therefore, it is clear that the set

$$\Gamma_a(x) = \{ \exists y \, ((y \; R^n \; x) \wedge \phi_n(y)) \}_{n \in \omega} \tag{6.106}$$

defines a up to automorphism.

The formula $\phi_n(x)$ are uniformly computable in the isomorphism types of the $\mathcal{S}_n$. It is clear from the definition of $\mathcal{E}^f$, that for all but finitely many $n \in \omega$,

$$\mathcal{S}_n \cong \begin{cases} \mathcal{E}_{n+\ell} & f(n+\ell)=1 \\ \mathcal{A}_{n+\ell} & \text{otherwise} \end{cases} \tag{6.107}$$

Let k be such that (6.107) holds for all $n \geq k$. So we can construct $\Gamma_a(x)$ given f and the results of the predicate

$$\mathfrak{M} \models (\chi^{\text{bnd}}_{=(m+\ell)}(b_m) \wedge \varphi_{\mathcal{E}_{(m+\ell)}}(b_m)) \tag{6.108}$$

for all $m < k$. As f is arithmetic and the formulae in (6.108) are all $\exists_{k+\ell}$, Γ_a is arithmetic.

In this proof, in order to define the automorphism type of a, we really only needed to know the isomorphism type of all of its ancestors. So we can extend this proof to a finite sequence $\overline{a}$ by specifying the ancestral relations between each of the elements of $\overline{a}$. □

Definition 6.7.14. $\mathfrak{A}$ is *arithmetically saturated* if for all $\overline{a} \in \mathfrak{A}$ and any arithmetic set of formulae $\Gamma(x, \overline{a})$, there is some $b \in \mathfrak{A}$ realizing Γ.

Corollary 6.7.15. $\mathfrak{M}^+$ *is arithmetically saturated.*

Proof. Take any $\overline{a} \in \mathfrak{M}$ and any arithmetic set of formulae $\Phi(x, \overline{a})$. By Theorem 6.7.13, we can assume without loss of generality that Γ is complete. Let $\mathfrak{B} \models \text{Th}(\mathfrak{M}, \overline{a})$ be any model with $b \in \mathfrak{B}$ realizing Φ.

It is clear from Proposition 6.7.2 that $\mathcal{T}_{\mathfrak{B}}$ embeds into $\mathcal{T}_{\mathfrak{M}}$. Furthermore, this embedding $f : \mathcal{T}_{\mathfrak{B}} \to \mathcal{T}_{\mathfrak{M}}$ is elementary; for each $\overline{c} \in \mathfrak{B}$ and $n \in \omega$, the strategy for the game $\text{EF}_n\big((\mathcal{T}_{\mathfrak{B}}, \overline{c}), (\mathcal{T}_{\mathfrak{M}}, f(c))\big)$ is just a modification of the one from Proposition 6.4.9. So if $b \in \mathcal{T}_{\mathfrak{B}}$, $\Phi(x, \overline{a})$ is realized in $\mathfrak{M}$.

Therefore, by Proposition 6.7.9, our only concern is that b may be in a copy of $\mathcal{E}^f$ for some nonarithmetic function f. Let Γ_b be the set defined in (6.106). It is clear from the proof of Theorem 6.7.13 that we can recover f from this set of formulae. As Φ is complete, $\Gamma \subseteq \Phi$. So f is arithmetic, since Φ is. □

Now that we know that $\mathfrak{M}^+$ is computably saturated, we can finally prove the result that motivated its construction.

Theorem 6.7.16. *Define*

$$\text{SAT}_{\text{Th}(\mathbb{N})}(e) =_{\text{def}} \mathfrak{M}_e \textit{ is arithmetically saturated} \tag{6.109}$$

Then $(\Sigma^0_\omega, \Pi^0_\omega) \leq_{1:1} (\overline{\text{SAT}_c}, \text{SAT}_{\text{Th}(\mathbb{N})})$.

Proof. We define $\mathcal{A}^+_{\omega+1}$ and $\mathcal{E}^+_{\omega+2}$ in the same way that we defined $\mathcal{E}^+_{\omega+1}$ and $\mathcal{A}^+_{\omega+2}$ in Definition 6.7.10. Let $\mathcal{P}(n)$ be a $\Pi^0_{\omega+2}$ predicate. By modifying the proof of Proposition 6.2.3, it is clear that we can construct a uniformly computable sequence of trees $\mathcal{T}_n$ such that

$$\mathcal{T}_n \cong \begin{cases} \mathcal{E}^+_{\omega+2} & \text{if } \neg\mathcal{P}(n) \\ \mathcal{A}^+_{\omega+2} & \text{otherwise} \end{cases} \tag{6.110}$$

We let $\mathfrak{A}_n$ be the disjoint union of $\mathcal{T}_n$ with infinitely many copies of Ω^*, and infinitely many $\mathcal{E}^f$ for each arithmetic f. Again, the structures $\mathfrak{A}_n$ are uniformly computable by the proof of Proposition 6.7.12. If $\mathcal{P}(n)$ holds, then $\mathfrak{A}_n \cong \mathfrak{M}^+$. Otherwise, the computable set of formulae in (6.77) that we used to show $\mathcal{E}_{\omega+2}$ is not computably saturated shows that $\mathfrak{A}_n$ is not computably saturated. □

Corollary 6.7.17. *Computable saturation is $\Pi^0_{\omega+2}$-complete.*

Corollary 6.7.18. *Arithmetic saturation is $\Pi^0_{\omega+2}$-complete.*

Proof. For each $m \in \omega$, let $\emptyset^m$ be the m^{th} iteration of the Turing jump on the empty set. A function is arithmetic if and only if it is equal to some $\{i\}^{\emptyset^m}$, for some i, $m \in \omega$. Hence, by replacing $\{i\}$ in (6.76) with $\{i\}^{\emptyset^m}$, we see that arithmetic saturation is $\Pi^0_{\omega+2}$. □

6.8 Σ_α Scott families

Atomic structures are nice, because as we saw in Theorem 2.4.2, any atomic structure $\mathfrak{A}$ can be elementarily embedded into any other model of $\mathrm{Th}(\mathfrak{A})$. The elementary embedding is constructed by a back-and-forth argument. Back-and-forth arguments are very desirable for computable structures, because when they can be done effectively, they produce computable maps between structures. In computable model theory, computable Σ_α Scott families are often used to make back-and-forth arguments effective.

Definition 6.8.1. A *computable Σ_α Scott family* for a structure $\mathfrak{A}$ is a finite set of parameters $\overline{c} \in \mathfrak{A}$ and a set of uniformly computable Σ_α formulae $\{\, \varphi_n \,\}_{n\in\omega}$ with the following properties.

(a) For all $\overline{a} \in \mathfrak{A}$, there is some $n \in \omega$ with $\mathfrak{A} \models \varphi_n(\overline{a}, \overline{c})$.

(b) For all $\overline{a}$, $\overline{b} \in \mathfrak{A}$ and $n \in \omega$, if $\mathfrak{A} \models (\varphi_n(\overline{a}, \overline{c}) \wedge \varphi_n(\overline{b}, \overline{c}))$, then there is an automorphism of $\mathfrak{A}$ taking $\overline{a}$ to $\overline{b}$.

We know from Chapter 16 of Ash and Knight [2] that, if a computable structure has a computable Σ_α Scott family, then it is Δ^0_α-categorical, though these properties are not necessarily identical. Hence we have some interest in determining the complexity of the predicate

$$\mathrm{SCOTT}_\alpha(e) =_{\mathrm{def}} \mathfrak{A}_e \text{ has a computable } \Sigma_\alpha \text{ Scott family} \tag{6.111}$$

If we could show that, for some fixed α, SCOTT_α and Δ^0_α-categoricity have different complexity, then this would prove that they are not identical. We will explore this point for the case $\alpha = 1$ in further detail in Chapter 7.

We know from Chapter 6 of Ash and Knight [2] that we can replace condition (b) with the following condition.

(b)′ For all $\overline{a}, \overline{b} \in \mathfrak{A}$, and $n \in \omega$, and computable Σ_α formula ψ, if $\mathfrak{A} \models (\varphi_n(\overline{a}, \overline{c}) \wedge \varphi_n(\overline{b}, \overline{c}))$, then

$$\mathfrak{A} \models \exists \overline{x}\, \psi(\overline{x}, \overline{a}, \overline{c}) \quad \Leftrightarrow \quad \mathfrak{A} \models \exists \overline{x}\, \psi(\overline{x}, \overline{b}, \overline{c}) \tag{6.112}$$

The proof that these definitions are equivalent is just the back-and-forth argument in Lemma 2.4.15. This means that

(6.113)

$$\begin{aligned} \mathrm{SCOTT}_\alpha(e) \Leftrightarrow \quad & \exists i, \overline{c}\, \forall \overline{a} \Big(\exists n \left(\{i\}(n) \downarrow= 1 \wedge \mathfrak{M}_e \models \phi^\alpha_n(\overline{a}, \overline{c}) \right) \wedge \\ & \forall k, \ell, \overline{b} \left(\left(\{i\}(k) \downarrow= 1 \wedge \mathfrak{M}_e \models \phi^\alpha_k(\overline{a}, \overline{c}) \wedge \mathfrak{M}_e \models \phi^\alpha_k(\overline{b}, \overline{c}) \right) \rightarrow \right. \\ & \left. \left(\mathfrak{M}_e \models \exists \overline{x}\, \phi^\alpha_\ell(\overline{x}, \overline{a}, \overline{c}) \leftrightarrow \mathfrak{M}_e \models \exists \overline{x}\, \phi^\alpha_\ell(\overline{x}, \overline{b}, \overline{c}) \right) \right) \Big) \end{aligned}$$

where ϕ^α_k is the uniform enumeration of all computable Σ_α formulae from section 6.3. It is clear that this predicate is at most $\Sigma^0_{\alpha+2}$. Once again, our goal of this section will be to show that this calculation is also a lower bound.

To do this, we will need to repeat much of the work that we did in section 6.4. Fortunately, we will be able to reuse most of the techniques that we developed there, albeit in a slightly different form. For the most part, we will be generalizing the Ehrenfeucht-Fraïssé games to infinitary sentences. For that, we need the following definition and result from Chapter 14 of Ash and Knight [2].

Definition 6.8.2. Let $\mathfrak{A}$, $\mathfrak{B}$ be structures in a fixed langauge. We define the relation $\precsim^\alpha$ by induction on α.

(a) $\mathfrak{A} \precsim^0 \mathfrak{B}$ if and only if $\mathfrak{A} \equiv_0 \mathfrak{B}$.

(b) For a successor ordinal $\alpha = \beta + 1$, $\mathfrak{A} \precsim^\alpha \mathfrak{B}$ if and only if for all $\overline{a} \in \mathfrak{A}$, there is some $\overline{b} \in \mathfrak{B}$ with $(\mathfrak{B}, \overline{b}) \precsim^\beta (\mathfrak{A}, \overline{a})$

(c) For a limit ordinal γ, $\mathfrak{A} \precsim^\gamma \mathfrak{B}$ if and only if $\mathfrak{A} \precsim^\beta \mathfrak{B}$ for all $\beta < \gamma$.

We write $\mathfrak{A} \sim^\alpha \mathfrak{B}$ whenever $\mathfrak{A} \precsim^\alpha \mathfrak{B}$ and $\mathfrak{B} \precsim^\alpha \mathfrak{A}$.

Theorem 6.8.3 (Ash and Knight [2]). *For all ordinals α, $\mathfrak{A} \precsim^\alpha \mathfrak{B}$ if and only if any Σ_α sentence true in $\mathfrak{B}$ is also true of $\mathfrak{A}$.*

Instead of playing Ehrenfeucht-Fraïssé games, we will directly use the definition of $\sim^\alpha$ to generalize Lemma 6.4.6 to infinitary sentences. In Lemma 6.4.6, we composed strategies on the limbs to construct a strategy for the entire tree. We need an analogue of this technique for $\sim^\alpha$.

Definition 6.8.4. Let $\mathcal{T}$, $\mathcal{S}$ be countable trees with $\overline{a} \in \mathcal{T}$, $\overline{b} \in \mathcal{S}$. Let $\{\, \mathcal{T}_n \,\}_{n\in\omega}$ enumerate the limbs attached to the root of $\mathcal{T}$, and similarly let $\{\, \mathcal{S}_n \,\}_{n\omega}$ enumerate the limbs attached to the root of $\mathcal{S}$. Let $\overline{a}|_n$ be the constants in $\overline{a}$ appearing in $\mathcal{T}_n$, and similarly define the constants $\overline{b}|_n$ for the limbs $\mathcal{S}_n$.

We say that $(\mathcal{T}, \overline{a}) \precsim^\alpha_{\text{limb}} (\mathcal{S}, \overline{b})$ if $\operatorname{depth}_{\mathcal{T}}(a_i) = \operatorname{depth}_{\mathcal{S}}(b_i)$ for all $i \leq |\overline{a}| = |\overline{b}|$, and there is a permutation $\pi_\alpha : \omega \to \omega$ such that,

(a) For all $i \leq k$, $a_i \in \overline{a}|_n$ if and only if $b_i \in \overline{b}|_{\pi_\alpha(i)}$

(b) For all $n \in \omega$,

$$(\mathcal{T}_n, \overline{a}|_n) \precsim^\alpha (\mathcal{S}_{\sigma(n)}, \overline{b}|_{\pi_\alpha(n)}) \tag{6.114}$$

Lemma 6.8.5. *Let $\mathcal{T}$, $\mathcal{S}$ be countable trees with $\overline{a} \in \mathcal{T}$, $\overline{b} \in \mathcal{S}$. For all ordinals α, if $(\mathcal{T}, \overline{a}) \precsim^\alpha_{\text{limb}} (\mathcal{S}, \overline{b})$ then $(\mathcal{T}, \overline{a}) \precsim^\alpha (\mathcal{S}, \overline{b})$.*

Proof. As usual, we proceed by induction on α.

Case $\alpha = 0$: We need only show that for all i, $j \leq k$,

$$\mathcal{T} \vDash (a_i \; R \; a_j) \quad \Leftrightarrow \quad \mathcal{T} \vDash (b_i \; R \; b_j) \tag{6.115}$$

Take any such i and j. Note that

$$\mathcal{T} \vDash (a_i \; R \; a_j) \quad \Leftrightarrow \quad \begin{cases} \mathcal{T}_n \vDash (a_i \; R \; a_j) & a_i,\, a_j \in \overline{a}|_n \\ \operatorname{depth}(a_i) = 0, \operatorname{depth}(a_j) = 1 & \text{otherwise} \end{cases} \tag{6.116}$$

And analogous result holds for b_i, b_j and $\mathcal{S}$. Therefore, (6.115) follows from and (6.114) and the fact that $\operatorname{depth}_{\mathcal{T}}(a_i) = \operatorname{depth}_{\mathcal{S}}(b_i)$ for all $i \leq |\overline{a}| = |\overline{b}|$.

Case $\alpha = \beta + 1$: Take any $\overline{c} \in \mathcal{T}$. We construct $\overline{d} \in \mathcal{S}$ as follows. If c_i is the root element of $\mathcal{T}$, we let d_i be the root element of $\mathcal{S}$. Otherwise, let $\overline{c}|_n$ be the constants in $\overline{c}$ appearing in $\mathcal{T}_n$. By (6.114) and the definition of $\precsim^\alpha$, for each $n \in \omega$, there is some collection of constants $\overline{e} \in \mathcal{S}_{\sigma(n)}$ such that

$$(\mathcal{S}_{\sigma(n)}, \overline{b}|_{\sigma(n)}, \overline{e}) \precsim^\beta (\mathcal{T}_i, \overline{a}|_n, \overline{c}|_n) \tag{6.117}$$

We let $\overline{d}|_{\sigma(n)} = \overline{e}$.

By our choice of $\overline{d}$, we have that for all $n \in \omega$,

$$(\mathcal{S}_{\sigma(i)}, (\overline{b}, \overline{d})|_{\sigma(n)}) \precsim^\beta (\mathcal{T}_i, (\overline{a}, \overline{c})|_n) \tag{6.118}$$

So $(\mathcal{S}, \overline{b}, \overline{d}) \precsim^\beta (\mathcal{T}, \overline{a}, \overline{c})$ by induction. As $\overline{c} \in \mathcal{T}$ was arbitrary, $(\mathcal{T}, \overline{a}) \precsim^\alpha (\mathcal{S}, \overline{b})$ follows from the definition of $\precsim^\alpha$.

Case α is a limit ordinal: Take any $\beta < \alpha$. Since (6.114) holds for α, a limit ordinal, it must also hold for β. Therefore, $(\mathcal{T}, \overline{a}) \precsim^\beta (\mathcal{S}, \overline{b})$ by induction. Hence $(\mathcal{T}, \overline{a}) \precsim^\alpha (\mathcal{S}, \overline{b})$ follows from the definition of $\precsim^\alpha$. □

Given this result, we can now modify our proof of Lemma 6.4.6 to get the following result for infinitary sentences.

Lemma 6.8.6. *Let α be a computable ordinal*

(a) If α is a successor ordinal, then for all $\overline{a} \in \mathcal{A}_\alpha$, there is some $\overline{b} \in \mathcal{E}_\alpha$ with $\mathrm{depth}(\overline{a}) = \mathrm{depth}(\overline{b})$ *such that $(\mathcal{A}_\alpha, \overline{a}) \sim^{\alpha-1} (\mathcal{E}_\alpha, \overline{b})$.*

(b) If α is a limit ordinal, then for all $n \in \omega$, $\mathcal{L}^\alpha_n \sim^{\alpha_n} \mathcal{L}^\alpha_\infty$.

Proof. As usual, we proceed by induction on α.

Case $\alpha = 1$: Clearly the constants $\overline{a}$ must be the single node of $\mathcal{A}_k$. Let $\overline{b}$ all be the root node of $\mathcal{E}_k$. Then, for all i, $j \leq |\overline{a}|$

$$\mathcal{A}_0 \vDash \neg a_i \; R \; a_j \quad \text{and} \quad \mathcal{E}_0 \vDash \neg b_i \; R \; b_j \tag{6.119}$$

Hence $(\mathcal{A}_1, \overline{a}) \equiv_0 (\mathcal{E}_1, \overline{b})$, and therefore $(\mathcal{A}_1, \overline{a}) \sim^0 (\mathcal{E}_1, \overline{b})$.

Case $\alpha = \beta + 1$, β a successor ordinal: Take any $\overline{a} \in \mathcal{A}_\alpha$. Let $\mathcal{T}_n$ be an enumeration of the limbs attached to the root of $\mathcal{A}_\alpha$. We let $\mathcal{S}_n$ be an enumeration of the limbs attached to the root of $\mathcal{E}_\alpha$, such that for all $n \in \omega$,

$$\mathcal{S}_n \cong \begin{cases} \mathcal{E}_\beta & n \text{ is even} \\ \mathcal{A}_\beta & \text{otherwise} \end{cases} \tag{6.120}$$

We as $\overline{a}$ is finite, we define a permutation $\pi_\alpha : \omega \to \omega$ such that $\pi(n)$ is even whenever $\overline{a}|_n$ is not empty. We choose $\overline{b} \in \mathcal{E}_\alpha$ to satisfy the requirements of Definition 6.8.4 where

$$(\mathcal{E}_\beta, \overline{a}|_n) \cong (\mathcal{E}_\beta, \overline{b}|_n) \tag{6.121}$$

or all $\overline{a}|_n$ not empty. We need to show that $(\mathcal{A}_\alpha, \overline{a}) \sim^\beta (\mathcal{E}_\alpha, \overline{b})$.

We will show that $(\mathcal{A}_\alpha, \overline{a}) \precsim^\beta (\mathcal{E}_\alpha, \overline{b})$; the proof that $(\mathcal{E}_\alpha, \overline{b}) \precsim^\beta (\mathcal{A}_\alpha, \overline{a})$ is analogous. Take any $\overline{c} \in \mathcal{E}_\alpha$, and any $n \in \omega$. If $\mathcal{S}_{\pi_\alpha(n)} \cong \mathcal{E}_beta$, then by our choice of elements in (6.121) we can choose some elements $\overline{e} \in \mathcal{T}_n$ with $\mathrm{depth}(\overline{e}) = \mathrm{depth}(\overline{c}|_n)$ such that

$$(\mathcal{S}_{\pi_\alpha(n)}, (\overline{b}, \overline{c})|_n \cong (\mathcal{T}_n, \overline{a}|_n, \overline{e}) \tag{6.122}$$

Otherwise, $\overline{b}_n = \emptyset$, and so by induction there is some $\overline{e} \in \mathcal{T}_n$ with such that $\operatorname{depth}(\overline{e}) = \operatorname{depth}(\overline{c}|_n)$ such that

$$(\mathcal{S}_{\pi_\alpha(n)}, (\overline{c})|_n \precsim^{\beta-1} (\mathcal{T}_n, \overline{e}) \tag{6.123}$$

Either way, we can choose $\overline{d} \in \mathcal{E}_\alpha$ to satisfy the requirements of Definition 6.8.4 for $\beta - 1$. In particular,

$$(\mathcal{S}_{\pi_\alpha(n)}, (\overline{b}, \overline{c})|_n \precsim^{\beta-1} (\mathcal{T}_n, (\overline{a}, \overline{d})|_n) \tag{6.124}$$

By Lemma 6.8.5, we have that $(\mathcal{E}_\alpha, \overline{b}, \overline{c}) \precsim^{\beta-1} (\mathcal{A}_\alpha, \overline{a}, \overline{d})$. So $(\mathcal{A}_\alpha, \overline{a}) \precsim^{\beta} (\mathcal{E}_\alpha, \overline{b})$ by the arbitrariness of $\overline{c} \in \mathcal{E}_\alpha$ and the definition of $\precsim^\beta$.

Case α a limit ordinal: Let $\mathcal{T}_k$ be an enumeration of the limbs of $\mathcal{L}^\alpha_n$ such that

$$\mathcal{T}_k \cong \begin{cases} \mathcal{A}_{\alpha_k} & k \leq n \\ \mathcal{E}_{\alpha_k} & \text{otherwise} \end{cases} \tag{6.125}$$

Let $\mathcal{S}_k$ be a similar enumeration of the limbs of $\mathcal{L}^\alpha_\infty$. By definition, we have that $\mathcal{T}_k \cong \mathcal{S}_k$ for all $k \leq n$. By induction, we have that $\mathcal{T}_k \sim^{\alpha_k - 1} \mathcal{S}_k$ for all $k > n$, and hence $\mathcal{T}_k \sim^{\alpha_n} \mathcal{S}_k$ for all such k. So $\mathcal{L}^\alpha_n \sim^{\alpha_n} \mathcal{L}^\alpha_\infty$ by Lemma 6.8.5.

Case $\alpha = \beta+1$, β a limit ordinal: Take any $\overline{a} \in \mathcal{A}_\alpha$. Let $\mathcal{T}_n$ be an enumeration of the limbs attached to the root of $\mathcal{A}_\alpha$ such that $\overline{a}|_k$ is empty for all $n < k$. We let $\mathcal{S}_n$ be an enumeration of the limbs attached to the root of $\mathcal{E}_\alpha$, such that $\mathcal{S}_n \cong \mathcal{T}_n$ for all $n \leq k$. Then we can choose $\overline{b} \in \mathcal{E}_\alpha$ to satisfy the requirements of Definition 6.8.4 where

$$(\mathcal{T}_n, \overline{a}|_n) \cong (\mathcal{S}_n, \overline{b}|_n) \tag{6.126}$$

for all $n \leq k$. We need to show that $(\mathcal{A}_\alpha, \overline{a}) \sim^\delta (\mathcal{E}_\alpha, \overline{b})$, for all $\delta < \beta$.

Fix any $\delta < \beta$. By induction, we can define a permutation $\pi_\alpha : \omega \to \omega$ such that $p(n) = n$ for all $n \leq k$ and

$$\mathcal{S}_{\pi_\alpha(n)} \cong \mathcal{L}^\alpha_\infty \quad \Rightarrow \quad \mathcal{S}_{\pi_\alpha(n)} \sim^{\delta+1} \mathcal{T}_n \tag{6.127}$$

So as in the case where β is a successor ordinal, we can use Lemma 6.8.5 to show that $(\mathcal{A}_\alpha, \overline{a}) \sim^\delta (\mathcal{E}_\alpha, \overline{b})$. □

Note that Lemma 6.8.6 implies both Lemma 6.4.6 and Lemma 6.4.7. Therefore, we could have proved this result instead in section 6.4 when we were calculating the complexity of homogeneity. We did not because the proofs in section 6.4 are more intuitive, and those proofs make the proof of Lemma 6.8.6 easier to understand.

Note that claim (a) of Lemma 6.8.6 implies that $\mathcal{A}_\alpha \precsim^\alpha \mathcal{E}_\alpha$ for all computable α. This fact will allow us to prove the main result of this section.

Theorem 6.8.7. *For all successor ordinals α, SCOTT_α is $\Sigma^0_{\alpha+2}$-complete.*

Proof. Let $\mathcal{P}$ be a $\Sigma^0_{\alpha+2}$ predicate, with $\mathcal{P}(n) = \exists x\, \forall y\, \mathcal{Q}(x, y, n)$ for a Σ^0_α predicate $\mathcal{Q}$. We recursively define the predicate.

$$(6.128) \qquad \mathcal{S}(x, y, n) \quad \Leftrightarrow \quad \begin{cases} \textbf{True} & \text{if } x,\, y = -1 \\ \mathcal{S}(x-1, y-1, n) \vee \bigwedge_{j=1}^{y} \mathcal{Q}(x, j, n) & \text{otherwise} \end{cases}$$

This predicate is very much like the one we defined in (6.58). We have just removed the variable x from the predicate in (6.58) and replaced y for i and x for y. Adapting the proof of Lemma 6.5.6, we get the following result.

Lemma 6.8.8. *For any x, $n \in \omega$, there is some $m \in \omega \cup \{\infty\}$ such that*

$$(6.129) \qquad \mathcal{S}(x, y, n) \quad \Leftrightarrow \quad y < m$$

Moreover, $m = \infty$ if and only if $\forall y\, \mathcal{Q}(z, y, n)$ for some $z \leq x$.

Let $\mathfrak{A}^k_\alpha$, $\mathfrak{E}^k_\alpha$ be the k-tagged intrepretations of $\mathcal{A}_\alpha$ and $\mathcal{E}_\alpha$ from Definition 6.3.6. As $\mathcal{S}(x, y, n)$, we construct a computable sequence of structures $\mathfrak{S}^{k,y}_n$ such that

$$(6.130) \qquad \mathfrak{S}^{k,y}_n \cong \begin{cases} \mathfrak{E}^k_\alpha & \text{if } y = 0 \text{ or } \mathcal{S}(k, y+1, n) \\ \mathfrak{A}^k_\alpha & \text{otherwise} \end{cases}$$

We let $\mathfrak{S}_n$ be the disjoint union of the structures $\mathfrak{S}^{k,y}_n$ for x, $y \in \omega$. These structures are illustrated in Figure 6.12.

Our claim is immediate from the following lemma.

Lemma 6.8.9. *For all $n \in \omega$, $\mathfrak{S}_n$ has a computable Σ_α Scott family if and only if $\mathcal{P}(n)$ holds.*

Proof. ($\Rightarrow$) Suppose the $\mathcal{P}(n)$ does not hold. Then, $\forall x \exists y\, \neg\mathcal{Q}(x, y, n)$. Hence it follows from Lemma 6.8.8 that $\forall x \exists y\, \neg\mathcal{S}(x, y, n)$. This means that for each $k \in \omega$, there is some $y \in \omega$ such that $\mathfrak{S}^{k,y}_n \cong \mathfrak{A}^k_\alpha$. So, for each $k \in \omega$, $\mathfrak{S}_n$ contains both a copy of $\mathfrak{A}^k_\alpha$ and $\mathfrak{E}^k_\alpha$.

Let $\overline{c} \in \mathfrak{S}$ be any sequence of elements, and let $k \in \omega$ be such that the $\overline{c}$ are not contained in any of the k-tagged components. Let a_k be the k-handle of some copy of $\mathfrak{A}^k_\alpha$ and b_k the k-handle of some copy of $\mathfrak{E}^k_\alpha$. Modifying the proof of Lemma 6.8.6 to include the k-handles, we have that $(\mathfrak{A}^k_\alpha, a_k) \precsim^\alpha (\mathfrak{E}^k_\alpha, b_k)$.

As a_k and b_k are not in $\overline{c}$ by our choice of k, it is easy to see that $(\mathfrak{S}_n, a_k, \overline{c}) \precsim^\alpha (\mathfrak{S}_n, b_k, \overline{c})$. So by Theorem 6.8.3, for any computable Σ_α formula with $\varphi(x, \overline{c})$ such that $\mathfrak{S}_n \models \varphi(a_k, \overline{c})$, we have that $\mathfrak{S}_n \models \varphi(b_k, \overline{c})$. So there is no such formula defining a_k up to automorphism in $\mathfrak{S}_n$.

($\Leftarrow$) Suppose the $\mathcal{P}(n)$. Then $\exists x \forall y\, \mathcal{Q}(x, y, n)$. Hence it follows from Lemma 6.8.8 that there is some $m \in \omega$ such that $\forall y\, \mathcal{S}(k, y, n)$ for all $k \geq m$. Furthermore, for each $k < m$, there are only finitely many y such that $\mathcal{S}(k, y, n)$. This means that $\mathfrak{D}_n$ has at most finitely many copies of $\mathfrak{E}^k_\alpha$ for $k < m$ and no copies of $\mathfrak{A}^k_\alpha$ for $k \geq m$. This structure is the one illustrated in Figure 6.12.

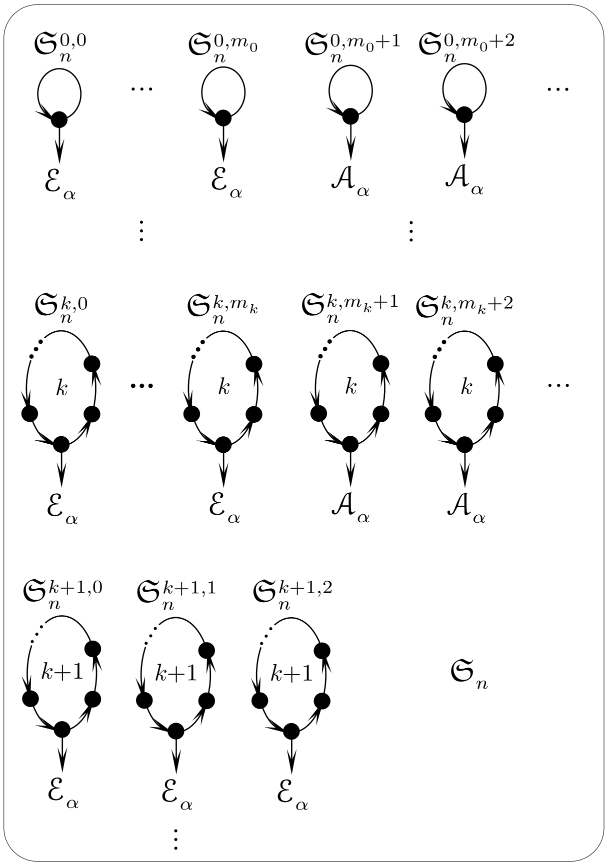

Figure 6.12: The structure $\mathfrak{S}_n$

By adapting the proof of Lemma 6.4.12 to the infinitary sentences defined in Lemma 6.3.4, we see that for each element of $\mathfrak{A}^k_\alpha$, there is a computable Σ_α formula that defines that element up to automorphism inside $\mathfrak{A}^k_\alpha$. Furthermore, for any sequence of elements in $\mathfrak{A}^k_\alpha$, we can define them up to automorphism by specifying their ancestral relations to each other, as we did in Theorem 6.7.13.

A similar result holds for $\mathfrak{E}^k_\alpha$. Furthermore the collection of all of these formulae is computable. So we need only show that we can define the k-handles of the components of $\mathfrak{S}_n$ up to automorphism by some uniform collection of computable Σ_α formulae with fixed parameters.

Let $\overline{c}$ be the k-handles of the finitely many copies of $\mathfrak{E}^k_\alpha$ for $k < m$. Then, for any k handle a_k which is not one of the elements $\overline{c}$, a_k is defined up to automorphism by the formula

$$\varphi(x, \overline{c}) =_{\text{def}} (x \; R^{k+1} \; x) \wedge \bigwedge_{i \leq |\overline{c}|} (x \neq c_i) \tag{6.131}$$

This is an existential formula. So $\mathfrak{S}_n$ has a computable Σ_α Scott family. ■

□

The case for when α is a limit ordinal is still unknown. The lack of uniformity in the structures $\mathcal{L}_m^\alpha$ prevents us from using the same proof that we did for successor ordinals. We do not know that $\mathcal{L}_\infty^\alpha \precsim^\alpha \mathcal{L}_m^\alpha$; we only know that $\mathcal{L}_\infty^\alpha \precsim^{\alpha_m} \mathcal{L}_m^\alpha$.

6.9 Applications to descriptive set theory

As with many of the properties in Chapter 5, none of the properties investigated in this chapter are unique to computable structures. So again it makes sense to look at the complexity of these properties with respect to countable structures.

We can effectively convert trees of the form $\mathcal{T} \subseteq \omega^\omega$ into the R-trees of this chapter in the obvious way. Therefore, we can construct boldface equivalents of all of the reductions in this chapter with the following result.

Proposition 6.9.1 (Boldface Proposition 6.2.3). *Let $\underset{\sim}{\mathcal{P}}(f) = \mathcal{P}(f, g)$ with $\mathcal{P}$ a Σ_α^0 predicate and let $g \in \omega^\omega$ a parameter.*

(a) Suppose α is a successor ordinal. We can uniformly compute from a Σ_α^0 index for $\mathcal{P}$ an index e such that, for all $f \in \omega^\omega$, $\{e\}^{f\oplus g}$ is the characteristic function of a tree $\mathcal{T}_f \subseteq \omega^{<\omega}$ with

$$\mathcal{T}_f \cong \begin{cases} \mathcal{E}_\beta & \text{if } \mathcal{P}(f) \\ \mathcal{A}_\beta & \text{otherwise} \end{cases} \tag{6.132}$$

(b) Suppose α is a limit ordinal. We can uniformly compute from a Σ_α^0 index for $\mathcal{P}$ an index e such that, for all $f \in \omega^\omega$, $\{e\}^{f\oplus g}$ is the characteristic function of a tree $\mathcal{T}_f \subseteq \omega^{<\omega}$ with

$$\mathcal{T}_f \cong \begin{cases} \mathcal{L}_\infty^\alpha & \text{if } \neg P(f) \\ \mathcal{L}_k^\alpha \text{ for some } k & \text{otherwise} \end{cases} \tag{6.133}$$

Proof. As before, we proceed by simultaneous effective transfinite recursion.

Case $\alpha = 1$: Then $\mathcal{P}(f, g) = \exists x\, \mathcal{Q}(x, f, g)$ for some computable predicate $\mathcal{Q}$. For all f, our function will put the empty sequence $\langle\rangle$ in $\mathcal{T}_f$ to represent the root node, and keep out any sequence of more than one element. For any sequence $|\sigma| = 1$, $\{e\}$ uses $f \oplus g$ as an oracle to search for a witness less than $\sigma(0)$ of the predicate $\mathcal{Q}(x, f, g)$. Our function puts $\langle n\rangle \in \mathcal{T}_f$ if and only if such a witness is found.

Case $\alpha = \beta + 1$, with β a successor ordinal: Then $\mathcal{P}(f, g) = \exists x\, \mathcal{Q}(x, f, g)$ for some Π_β^0 predicate $\mathcal{Q}(x, f, g)$. As before, we use our inductive hypothesis to give ourselves a uniformly computable sequence e_n such that $\{\, e_n \,\}^{f\oplus g} = \mathcal{T}_{n,f}$ satisfies (6.132) for $\mathcal{Q}$. Let $\{\, i \,\} = \mathcal{E}_\beta$ be computable. Then

$$\{e\}^{f\oplus g}(\tau) = \begin{cases} 1 & \text{if } |\tau| = 0 \\ \{\, e_n \,\}^{f\oplus g}(\sigma) & \text{if } \tau = (2\langle n, m\rangle)^\frown\sigma \\ \{\, i \,\}(\sigma) & \text{if } \tau = (2n+1)^\frown\sigma \end{cases} \tag{6.134}$$

It is clear that this definition works.

Case α is a limit ordinal: Again we can assume with out loss of generality that $P(f)$ is a computable disjunction of $Q_k(f) \in \Sigma^0_{\alpha_k}$. By induction, we have a computable sequence e_n satisfying (6.132) for the predicate

$$\mathcal{C}(m,n) =_{\text{def}} \bigvee_{k \leq m} \mathcal{Q}_k(n) \tag{6.135}$$

We define

$$\{e\}^{f\oplus g}(\tau) = \begin{cases} 1 & \text{if } |\tau| = 0 \\ \{\, e_n \,\}^{f\oplus g}(\sigma) & \text{if } \tau = (n)^\frown\sigma \end{cases} \tag{6.136}$$

It is clear that this definition works.

Case $\alpha = \beta + 1$, with β a limit ordinal: Again we write $\mathcal{P}(n) = \exists x\, \mathcal{Q}(x,n)$ for some Π^0_β predicate $\mathcal{Q}(x,n)$. By induction, there is a computable sequence of e_n such that $\{\, e_n \,\}^{f\oplus g} = \mathfrak{T}_{n,f}$ satisfies (6.133) for Q. For each $m \in \omega$, let $\{\, i_m \,\} \cong \mathcal{L}^\alpha_m$. Then

$$\{e\}^{f\oplus g}(\tau) = \begin{cases} 1 & \text{if } |\tau| = 0 \\ \{\, e_n \,\}^{f\oplus g}(\sigma) & \text{if } \tau = (2\langle n,m\rangle)^\frown\sigma \\ \{\, i_m \,\}(\sigma) & \text{if } \tau = (2n+1)^\frown\sigma \end{cases} \tag{6.137}$$

It is clear that this definition works. $\square$

Corollary 6.9.2. *Consider following model theoretic properties defined over the collection of countable structures of our language $\mathcal{L}$ with a single binary relation.*

(a) $\equiv$ *is* $\underset{\sim}{\Pi}^0_\omega$*-complete.*

(b) $\equiv^\alpha_c$ *is* $\underset{\sim}{\Pi}^0_{\alpha+1}$*-complete whenever α is a successor ordinal.*

(c) $\equiv^\alpha_c$ *is* Π^0_α*-complete whenever α is a limit ordinal.*

(d) Homogeneity is $\underset{\sim}{\Pi}^0_{\omega+2}$*-complete.*

(e) The class of atomic structures is $\underset{\sim}{\Pi}^0_{\omega+2}$*-complete.*

(f) Computable saturation is $\underset{\sim}{\Pi}^0_{\omega+2}$*-complete.*

(g) For successor ordinals α, the existence of a computable Σ_α Scott family is $\underset{\sim}{\Sigma}^0_{\alpha+2}$*-complete.*

Chapter 7

Computable Categoricity

7.1 Introduction

In Chapter 5, we examined the properties of hyperarithmetic stability and categoricity. If we let Γ in Definition 5.5.1 be the collection of all computable functions, then we get the notions of computable categoricity and stability.

Definition 7.1.1. A computable structure $\mathfrak{A}$ is *computably categorical* if, for all computable structures $\mathfrak{B} \cong \mathfrak{A}$, there is a computable isomorphism $f : \mathfrak{B} \cong \mathfrak{A}$. Similarly, $\mathfrak{A}$ is *computably stable* if, whenever $\mathfrak{B}$ is a computable structure isomorphic to $\mathfrak{A}$, all isomorphisms from $\mathfrak{B}$ to $\mathfrak{A}$ are computable.

As with hyperarithmetic categoricity and stability, these two properties seem to be Π^1_1. That is because they are defined by the predicates

$$\text{CAT}_c(e) =_{\text{def}} \forall f\, \forall i\, \exists j\, \big((f : \mathfrak{M}_e \cong \mathfrak{M}_i) \to (\{\, j\,\} : \mathfrak{M}_e \cong \mathfrak{M}_i)\big) \tag{7.1}$$

and

$$\text{STAB}_c(e) =_{\text{def}} \forall f\, \forall i\, \exists j\, \big((f : \mathfrak{M}_e \cong \mathfrak{M}_i) \to (\{\, j\,\} = f)\big) \tag{7.2}$$

Because of this fact, there has been much research into the existence of equivalent, computationally less complicated, equivalent definitions. In the case of computable categoricity, one possible candidate that has been well-studied is the existence of a computable Σ_1 Scott family, which we defined in Chapter 6. It is well known that the existence of a computable Scott family implies computable categoricity. Furthermore, Goncharov [16] has shown that, in the case of 2-decidable structures, they are equivalent.

Theorem 7.1.2 (Goncharov [16]). *Let $\mathfrak{A}$ be a 2-decidable structure. Then $\mathfrak{A}$ is computably categorical if and only if it has a computable Σ_1 Scott family.*

A similar candidate for computable stability is the existence of a computable Σ_1 defining family.

Definition 7.1.3. A Σ_α *defining family* for a structure $\mathfrak{A}$ is a finite set of parameters $\overline{c} \in \mathfrak{A}$ and a set Φ of Σ_α formulae with the following properties.

(a) For all $\overline{a} \in \mathfrak{A}$, there is some $\varphi_{\overline{a}} \in \Gamma$ such that $\mathfrak{A} \models \varphi_{\overline{a}}(\overline{a}, \overline{c})$.

(b) For all $\overline{a}$, $\overline{b} \in \mathfrak{A}$, if $\mathfrak{A} \models \varphi_{\overline{a}}(\overline{b}, \overline{c})$, then $\overline{a} = \overline{b}$.

A *computable* Σ_α *defining family* for $\mathfrak{A}$ is a set of uniformly computable Σ_α formulae $\{\, \varphi_n \,\}_{n\in\omega} = \Phi$ such that Φ is a Σ_α defining family.

It is clear from Definition 7.1.3 that any computable Σ_1 defining family is also a computable Σ_1 Scott family. However, these two notions are not equivalent; the existence of a computable Σ_1 defining family also implies that the structure is rigid. So while $(\mathbb{Q}, \leq)$, the countable dense linear ordering without endpoints, has a computable Σ_1 Scott family, it does not have a computable Σ_1 defining family.

Suppose that $\mathfrak{A}$ is a computable structure with a computable Σ_1 defining family. Then, as we pointed out, $\mathfrak{A}$ is rigid, and it has a computable Σ_1 Scott family. So the existence of a computable Σ_1 defining family implies that $\mathfrak{A}$ is computably stable. Again, Goncharov [16] has shown that, in the case of 1-decidable models, these two are equivalent.

Theorem 7.1.4 (Goncharov [16]). *Let $\mathfrak{A}$ be a 1-decidable structure. Then $\mathfrak{A}$ is computably stable if and only if it has a computable $\exists$ defining family.*

These results are all particularly nice because both the existence of a computable Σ_1 Scott family and the existence of a computable defining family are much less complex than Π^1_1. We have already seen from Theorem 6.8.7 that the existence of a computable Σ_1 Scott family is Σ^0_3-complete. A similar result holds for the existence of a computable defining family.

Theorem 7.1.5. *Define the predicate*

$$\mathrm{DEF}^c_1(e) =_{\mathrm{def}} \mathfrak{M}_e \textit{ has a computable } \Sigma^0_1 \textit{ defining family} \tag{7.3}$$

Then DEF^c_1 *is* Σ^0_3*-complete.*

Proof. Let ϕ_n be an enumeration of all the computable Σ_1 formulae in our language. Then, as in section 6.8, we can write DEF^c_1 as

(7.4)

$$\mathrm{DEF}^c_1(e) \Leftrightarrow \begin{array}{l} \exists i, \overline{c}\, \forall \overline{a} \Big[\exists n \left(\{i\}(n) \downarrow = 1 \wedge \mathfrak{M}_e \models \phi_n(\overline{a}, \overline{c}) \right) \wedge \\ \qquad \forall k, \ell\, \forall \overline{b} \Big(\left(\{i\}(k) \downarrow = 1 \wedge \mathfrak{M}_e \models \phi^\alpha_k(\overline{a}, \overline{c}) \wedge \phi^\alpha_k(\overline{b}, \overline{c}) \right) \to \overline{a} = \overline{b} \Big) \Big] \end{array}$$

Therefore, as any computable Σ_1 defining family is also a computable Σ_1 Scott family, we need only show that $(\Sigma^0_3, \Pi^0_3) \leq (\mathrm{DEF}^c_1, \overline{\mathrm{SCOTT}_1})$.

Let $\mathcal{P}(n)$ be any Σ^0_3 predicate. As usual, we will construct a uniformly computable sequence of structures $\mathfrak{A}_n$ such that $\mathfrak{A}_n$ has a computable Σ_1 defining family when $\mathcal{P}(n)$ holds, and no computable Σ_1 Scott family when it does not. To simplify the proof for the negative case, we need the following lemma.

Lemma 7.1.6. *Let $\mathfrak{A}$ be a computable structure with a computable Σ_1 Scott family. Then $\mathfrak{A}$ has a computable Σ_1 Scott family Φ where all of the formulae in Φ are $\exists$.*

Proof. Let Φ be a computable Σ_1 Scott family with parameters $\overline{c}$. It is enough to show that for any $\varphi \in \Phi$, there is an $\exists$ formula ψ such that

$$\mathfrak{A} \models \forall \overline{x} \left(\varphi(\overline{x}, \overline{c}) \leftrightarrow \psi(\overline{x}, \overline{c}) \right) \tag{7.5}$$

Take any $\varphi \in \Phi$ and let $\overline{a} \in \mathfrak{A}$ be such that $\mathfrak{A} \models \varphi(\overline{a}, \overline{c})$. As φ is a computable Σ_1 formula, there is a computable sequence $\{ \chi_k \}_{k \in \omega}$ of $\exists$ formulae such that

$$\mathfrak{A} \models \forall \overline{x} \left(\varphi(\overline{x}, \overline{c}) \leftrightarrow \bigvee_{k \in \omega} \psi_k(\overline{x}, \overline{c}) \right) \tag{7.6}$$

Let $k \in \omega$ be least such that $\mathfrak{A} \models \psi_k(\overline{a}, \overline{c})$. For any $\overline{b} \in \mathfrak{A}$, if $\mathfrak{A} \models \varphi(\overline{b}, \overline{c})$, then there is an automorphism taking $\overline{a}$ to $\overline{b}$. So, in this case $\mathfrak{A} \models \psi_k(\overline{b}, \overline{c})$. Therefore, ψ_k satisfies (7.5). ■

As usual, the structures $\mathfrak{A}_n$ in our reduction will be graphs with respect to the binary relation R. These graphs will be disjoint unions of tagged cycles.

Definition 7.1.7. A connected graph $\mathcal{G}$ is a *tagged cycle* if it is a cycle together with some additional element $a \in \mathcal{G}$ not in the cycle such that there is a unique element $b \in \mathcal{G}$ in the cycle with $(a\ R\ b)$, $(b\ R\ a)$. This element a is called the *tag* of the cycle, while b is the *base node* of the cycle. Tagged cycles may be infinite; in this case the graph is simply an infinite chain with a tag attached to a unique node in the chain. Figure 7.1 illustrates both finite and infinite tagged cycles.

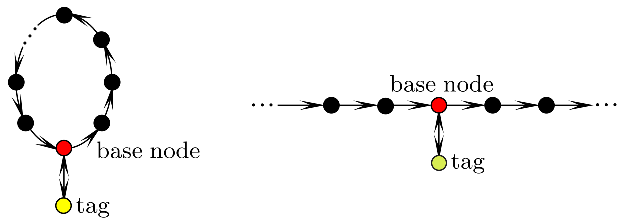

Figure 7.1: Finite and infinite tagged cycles

For $k \in \omega$, let $\mathcal{C}_k$ be the graph with exactly one finite tagged cycle of each length, and exactly k many infinite tagged cycles. Note that for any $m < n$, there is an existential formula specifying that an element is m steps away from the tag in a tagged cycle of length n. Therefore, all of the elements in the finite tagged cycles of $\mathfrak{A}$ have defining formulae without parameters. Furthermore, any element in an infinite tagged cycle can be uniquely identified by how many steps it is away from its tag. So, as $\mathcal{C}_k$ has only k many infinite tagged cycles, it has a Σ_1 defining family, where the parameters are the tags of the infinite cycles. Because all possible finite cycles appear in $\mathfrak{A}$, this Σ_1 defining family is clearly computable.

On the other hand, let $\mathcal{C}_\infty$ be the graph with exactly one finite tagged cycle of each length and infinitely many infinite cycles. Then Lemma 7.1.6 together with the following lemma shows that $\mathcal{C}_\infty$ has no computable Σ_1 Scott family.

Lemma 7.1.8. *Let $\overline{c} \in \mathcal{C}_\infty$ be finitely many parameters, and let $a \in \mathcal{C}_\infty$ be the tag of an infinite cycle not containing any of the elements of $\overline{c}$. For any $\exists$ formula φ such that $\mathcal{C}_\infty \vDash \varphi(a, \overline{c})$, there is a tag $b \in \mathcal{C}_\infty$ of a finite cycle such that $\mathcal{C}_\infty \vDash \varphi(b, \overline{c})$.*

Proof. Let $\varphi = \exists \overline{x}\, \psi(\overline{x})$, where ψ is quantifier free, and let $\overline{d} \in \mathcal{C}_\infty$ be such that $\mathcal{C}_\infty \vDash \psi(a, \overline{c}, \overline{d})$. Only finitely many elements of the cycle tagged by a can appear in $\overline{d}$; let n be the furthest distance any of these is away from a. Let b be the tag of a cycle of length $k \geq 2n + 1$ not containing any of the elements of $\overline{c}$ or $\overline{d}$.

We can assume without loss of generality that all of the elements of $\overline{d}$ are in the cycle tagged by a; in any elements of $\overline{d}$ that are not in this cycle can be included in $\overline{c}$. As all of the elements of $\overline{d}$ are less than n steps away from a, we can choose a sequence of constants $\overline{d}'$ in the cycle tagged by b such that

$$\langle a, \overline{c}, \overline{d} \rangle_{\mathcal{C}_\infty} \cong \langle b, \overline{c}, \overline{d}' \rangle_{\mathcal{C}_\infty} \tag{7.7}$$

We do this by choosing, for each $d_i \in \overline{d}$, a d_i' in the cycle tagged by b such that for all $m \leq n$,

$$\mathcal{C}_\infty \vDash (b\ R^m\ d_i' \leftrightarrow a\ R^m\ d_i) \quad \text{or} \quad \mathcal{C}_\infty \vDash (d_i'\ R^m\ b \leftrightarrow d_i\ R^m\ b) \tag{7.8}$$

Since the cycle tagged by b has length $k \geq 2n + 1$, we have that

$$\mathcal{C}_\infty \vDash (d_i\ R\ d_j \leftrightarrow d_i'\ R\ d_j') \tag{7.9}$$

for all d_i, $d_j \in \overline{d}$. This concept is illustrated in Figure 7.2.

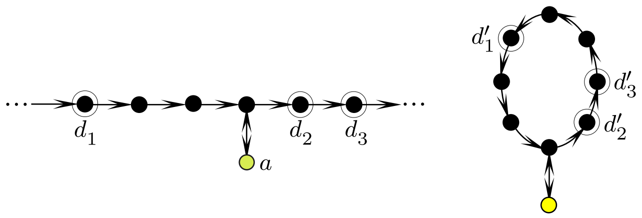

Figure 7.2: Choice of constants $\overline{d}'$

As ψ is quantifier free, (7.7) implies that $\mathcal{C}_\infty \vDash \psi(b, \overline{c}, \overline{d}')$. Hence we have that $\mathcal{C}_\infty \vDash \varphi(a, \overline{c})$. ∎

Given this, we need only construct our sequence $\mathfrak{A}_n$ so that

$$\mathfrak{A}_n \cong \begin{cases} \mathcal{C}_k \text{ for some } k \in \omega & \text{if } \mathcal{P}(n) \\ \mathcal{C}_\infty & \text{otherwise} \end{cases} \tag{7.10}$$

We write $\mathcal{P}(n) = \exists x \, \forall y \, \exists z \, \mathcal{Q}(x, y, z, n)$ where $\mathcal{Q}$ computable. We proceed in stages, constructing finite structures $\mathfrak{A}_n^s$ so that $\mathfrak{A}_n = \bigcup_s \mathfrak{A}_n^s$. At stage s, $\mathfrak{A}_n^s$ will be composed of several finite cycles, and several graphs which are attempts at an infinite cycle, as illustrated in Figure 7.3. Each potential infinite cycle will be "marked" by some element $x < s$.

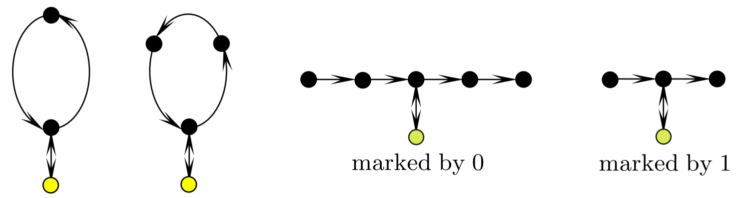

Figure 7.3: Stage s of the construction

The idea is that, if x is a witness for $\forall y \, \exists z \, \mathcal{Q}(x, y, z, n)$, then all the graphs marked with $w \geq x$ will eventually become finite cycles. In order to carry this out, we will have witness functions y_s, z_s. We say that x *requires attention* at stage s if $\mathcal{Q}(x, y_s(x), z_s(x), n)$ holds.

Stage $s = -1$:
We let $\mathfrak{A}_n^{-1}$ be the empty graph. Furthermore, for all x we define $y_{-1}(x) = z_{-1}(x) = 0$.

Stage $s = t + 1$:
If no $x < s$ requires attention, set $y_s(x) = y_{s-1}(x)$, $z_s(x) = z_{s-1}(x) + 1$ for all $x < s$. Furthermore, add to $\mathfrak{A}_n^s$ the least finite cycle not seen so far, and a new potential infinite cycle marked by s. Finally, increase the length of each marked graph by one.

If some $x < s$ requires attention, choose the least such x. For all potential infinite cycles in $\mathfrak{A}_n^s$ marked by $w > x$, we remove the mark, and close them off as finite cycles so that their lengths are unique in $\mathfrak{A}_n$. We now add new potential infinite cycles for each $w > x$, $w \leq s$, and mark them appropriately. We increase the length of each graph marked by $w \leq x$ by one. Finally, we add the least finite cycle not seen so far and define

(7.11)

$$y_s(w) = \begin{cases} y_{s-1}(w) & \text{if } w \leq x \\ y_{s-1}(w) + 1 & \text{if } w = x \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad z_s(w) = \begin{cases} z_{s-1}(w) + 1 & \text{if } w \leq x \\ 0 & \text{otherwise} \end{cases}$$

This completes our construction. Clearly $\mathfrak{A}_n = \bigcup_s \mathfrak{A}_n^s$ is computable; any element ξ of $\mathrm{diag}(\mathfrak{A}_n)$ is determined by $\mathfrak{A}^s$ for the least s such that $\mathfrak{A}_n^s$ contains the elements of ξ. Furthermore, our construction ensures that $\mathfrak{A}_n$ has exactly one finite cycle of each length. So, we need only show that it has a finite number of infinite cycles if and only if $\mathcal{P}(n)$.

It is clear from our construction that a marked potential infinite cycle becomes an infinite cycle in $\mathfrak{A}_n$ if and only if the mark is never removed. Therefore, if x requires attention infinitely often, then it has at most x infinite cycles. So $\mathfrak{A}_n$ has a finite number of infinite cycles whenever there is some $x \in \omega$ which receives attention infinitely often.

Suppose now that no x requires attention infinitely often. Then for each $x \in \omega$, there is some stage s such that the graph marked by x at stage s becomes an infinite cycle, namely the least stage such that all $w < x$ no longer require attention. So $\mathfrak{A}_n$ has a finite number of infinite cycles if and only if some $x \in \omega$ receives attention infinitely often. As it is clear from our construction that $\exists x \, \forall y \, \exists z \, \mathcal{Q}(x, y, z, n)$ if and only if there is some x which requires attention infinitely often, we are done. $\square$

While the existence of computable Scott families and defining families are computationally less complicated than computable categoricity and stability, they are not equivalent in general. Goncharov [16] has shown that there is a computably stable model with no computable defining family. Similarly, Kudinov [33] has shown that there is a computably categorical, 1-decidable model with no computable Scott family.

An alternate proof that they are not equivalent in general can be obtained by looking at the computational complexity of these two properties. If we can show that neither computable categoricity nor computable stability are Σ^0_3, then we know that are not equivalent to the existence of either a computable Σ_1 Scott family or a computable Σ_1 defining family, respectively. Therefore, our goal of this chapter will be to prove the following result.

Theorem 7.1.9. $(\Sigma^0_4, \Pi^0_4) \leq (\overline{\mathrm{CAT}_c}, \mathrm{STAB}_c)$.

The proof of this theorem is an infinite injury construction, and will take up the remainder of this chapter. In order to define this construction, we will fix, from now on, a Π^0_4 predicate $\mathcal{P}(n) = \forall x \, \exists y \, \forall z \, \mathcal{Q}(x, y, z, n)$, where $\mathcal{Q}$ is computable.

7.2 The priority tree

As in Theorem 7.1.5, the structures $\mathfrak{A}_n$ in our reduction will be a disjoint union of connected, finite graphs. These finite graphs will each be uniquely identifiable by an existential formula. Therefore, $\mathfrak{A}_n$ will have a defining family, though this family may not be computable. The existence of a defining family ensures that $\mathfrak{A}_n$ will be rigid. As any rigid, computably categorical model is computably stable, our construction only needs to guarantee the following.

(Π^0_4) If $\mathcal{P}(n)$ holds, our construction should provide computable isomorphisms from $\mathfrak{A}_n$ to each computable $\mathfrak{M}_e \cong \mathfrak{A}_n$.

(Σ^0_4) If $\mathcal{P}(n)$ does not hold, our construction should provide a computable structures $\mathfrak{B} \cong \mathfrak{A}_n$ such that no computable function is an isomorphism between them.

To do this, we have two types of requirements: positive and negative. The positive requirements will ensure that $\mathfrak{A}_n$ is computably categorical, while the negative requirements will provide a witness that it is not. These will be arranged on a priority tree so that we will only statisfy finitely many of the requirements of the wrong type, depending on $\mathcal{P}(n)$.

To satisfy (Π^0_4), for each element $\sigma \in \mathcal{T}$ of our priority tree $\mathcal{T}$, and $x \in \omega$, we will try to build a computable function $f_{x,\sigma} : \mathfrak{M}_x \to \mathfrak{A}_n$ which is an isomorphism provided that $\mathfrak{M}_x \cong \mathfrak{A}_n$. Our positive requirements then become

$$(P_{x,\sigma}) \qquad\qquad \mathfrak{M}_x \cong \mathfrak{A}_n \quad \Rightarrow \quad f_{x,\sigma} : \mathfrak{M}_x \cong \mathfrak{A}_n$$

If, for all $x \in \omega$, we statisfy $(P_{x,\sigma})$ for some $\sigma \in \mathcal{T}$, then $\mathfrak{A}_n$ is computably categorical.

To satisfy (Σ^0_4), for each $\sigma \in \mathcal{T}$ and $y \in \omega$, we will try to build a computable model $\mathfrak{B}_\sigma \cong \mathfrak{A}_n$ so that $\{\, y \,\}$ is not an isomorphism between $\mathfrak{B}_\sigma$ and $\mathfrak{A}_n$. Our negative requirements then become

$$(N_{\sigma,y}) \qquad \mathfrak{B}_\sigma \cong \mathfrak{A}_n \quad \text{and} \quad \{\, y \,\} : \mathfrak{B}_\sigma \to \mathfrak{A}_n \quad \text{is not an isomorphism}$$

If there is some $\sigma \in \mathcal{T}$ in our priority tree such that we satisfy $(N_{\sigma,y})$ for all $y \in \omega$, then $\mathfrak{A}_n$ is not computably categorical.

Our priority tree $\mathcal{T}$ will be finite sequences over the set $\{\, \Pi, \Sigma, \infty, f, m, u \,\}$. Furthermore, we define a function $\lceil\cdot\rceil$ from $\mathcal{T}$ to the union of the set of requirements, $(P_{x,\sigma})$ and $(N_{\sigma,y})$, and the set of all pairs $\langle x, y\rangle$. This function $\lceil\cdot\rceil$ will determine what requirements are being satisfied by the current approximation of the true path. We define $\mathcal{T}$ and $\lceil\cdot\rceil$ by simultaneous induction.

(a) The root of $\mathcal{T}$ is the empty sequence $\langle\rangle \in \mathcal{T}$ with $\lceil\langle\rangle\rceil = \langle 0, 0\rangle$.

(b) Whenever $\sigma \in \mathcal{T}$ with $\lceil\sigma\rceil = \langle x, y\rangle$, we have that $\sigma^\frown\Pi \in \mathcal{T}$ with $\lceil\sigma^\frown\Pi\rceil = P_{x,\sigma}$.

(c) Whenever $\sigma \in \mathcal{T}$ with $\lceil\sigma\rceil = \langle x, y\rangle$, we have that $\sigma^\frown\Sigma \in \mathcal{T}$. Furthermore, $\lceil\sigma^\frown\Sigma\rceil = N_{\tau,y}$ where $\tau \preceq \sigma$ (i.e. $\sigma = \tau^\frown\rho$ for some ρ) is such that $\lceil\tau\rceil = \langle x, 0\rangle$.

(d) Whenever $\sigma \in \mathcal{T}$ with $\lceil\sigma\rceil = P_{x,\rho}$, we have that $\sigma^\frown\infty$, $\sigma^\frown f \in \mathcal{T}$ with $\lceil\sigma^\frown\infty\rceil = \lceil\sigma^\frown f\rceil = \langle x+1, 0\rangle$.

(e) Whenever $\sigma \in \mathcal{T}$ with $\lceil\sigma\rceil = N_{\rho,y}$, we have that $\sigma^\frown m$, $\sigma^\frown u \in \mathcal{T}$ with $\lceil\sigma^\frown m\rceil = \lceil\sigma^\frown u\rceil = \langle x, y+1\rangle$.

Because of definition (c), there is some question as to whether $\mathcal{T}$ is well-defined. However, this can be resolved by a simple induction. In fact, all of the following properties of our tree are immediate from the inductive definition of $\mathcal{T}$ and $\lceil\cdot\rceil$.

Lemma 7.2.1. *For any $\sigma \in \mathcal{T}$, the following all hold.*

(a) If $\lceil\sigma\rceil = \langle x, y\rangle$, there is a unique $\rho \preceq \sigma$ with $\lceil\rho\rceil = \langle x, 0\rangle$.

(b) Let τ be the greatest $\tau \preceq \sigma$ corresponding to a positive requirement $(P_{x,\alpha})$. Then for all $\tau \prec \rho \preceq \sigma$, we have that $\lceil\rho\rceil = \langle x+1, y\rangle$ or $\lceil\rho\rceil = N_{\beta,y}$ for $\lceil\beta\rceil = \langle x+1, 0\rangle$. Furthermore, these ρ range over all such $y < \frac{|\sigma|-|\tau|}{2}$.

(c) If $\lceil\sigma\rceil = P_{x,\rho}$, then for all $w < x$ there is some $\tau \preceq \sigma$ with $\lceil\tau\rceil = \langle w, 0\rangle$.

(d) For all $i < |\sigma|$,

$$\sigma(i) \in \begin{cases} \{\,\Pi, \Sigma\,\} & \text{if } i = 2k \\ \{\,\infty, f\,\} & \text{if } i = 2k+1,\ \lceil\sigma\rceil = P_{x,\rho} \\ \{\,m, u\,\} & \text{if } i = 2k+1,\ \lceil\sigma\rceil = N_{\rho,y} \end{cases} \tag{7.12}$$

The ordering $<_L$ on $\mathcal{T}$ is the lexicographical ordering induced by $\Pi < \Sigma$, $\infty < f$, and $m < u$. This gives us the priority tree illustrated in Figure 7.4.

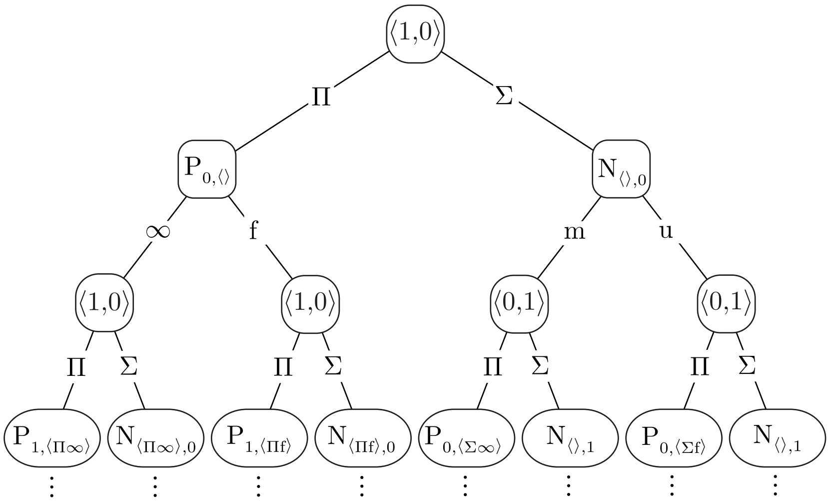

Figure 7.4: The priority tree

The motivation behind this tree is as follows. Nodes $\sigma \in \mathcal{T}$ with $\lceil\sigma\rceil = \langle x, y\rangle$ attempt to determine whether or not $\mathcal{P}(n)$ holds by approximating the predicate $\forall z\, \exists w\, \mathcal{Q}(x, y, z, w, n)$ on $\langle x, y\rangle$. The node $\sigma^\frown\Pi$ corresponds to the positive outcome of this approximation, while $\sigma^\frown\Sigma$ corresponds to the negative outcome.

Nodes $\sigma \in \mathcal{T}$ with $\lceil\sigma\rceil = P_{x,\rho}$ attempt to satisfy the associated positive requirement. The node $\sigma^\frown\infty$ corresponds to the case in which $\mathfrak{M}_x \cong \mathfrak{A}_n$, while $\sigma^\frown f$ corresponds to the case $\mathfrak{M}_x \ncong \mathfrak{A}_n$.

Finally, nodes $\sigma \in \mathcal{T}$ with $\lceil \sigma \rceil = N_{\rho,y}$ attempt to satisfy the associated negative requirement. The node $\sigma^\frown m$ corresponds to the case in which that requirement has been met, while $\sigma^\frown u$ is the case in which it is still not met.

7.3 Satisfying the requirements

To satisfy $(N_{\sigma,y})$, we simply need some element of our model $\mathfrak{A}_n$ witnessing that $\{y\}$ is not an isomorphism from $\mathfrak{B}_\sigma$ to $\mathfrak{A}_n$. Recall that $\mathfrak{A}_n$ (and hence $\mathfrak{B}_\sigma$) will be made of finite, connected graphs that are uniquely identified by an existential formula. So for any $b \in \mathfrak{B}_\sigma$, our only concern is when $\{y\}$ maps b to the corresponding element in the corresponding component of $\mathfrak{D}$; otherwise $\{y\}$ is not an isomorphism.

Suppose this does happen for some $b \in \mathfrak{B}_\sigma$. In this case, we will extend our graph to a new one which is also uniquely definable by an existential formula. Furthermore, we will chose our extension so the embedding of the old graph into this extension will not be unique. This will allow us to extend the graph differently in both $\mathfrak{A}_n$ and $\mathfrak{B}_\sigma$. That way we can ensure that $\{y\}(b)$ is definably different from b, and so b witnesses that $\{y\}$ is not an isomorphism.

To do this, we will use techniques similar to those introduced in Khoussainov and Shore [31]. The finite, connected graphs in our model will be *k-modules*. There will be multiple types of k-modules, but $\mathfrak{A}_n$ will have at most one k-module for any $k \in \omega$. The structure of these k-modules will ensure that our model has a defining family. However, this family will not necessarily be computable because we may not have an algorithm to determine which particular type of k-module our model has for each k.

Definition 7.3.1. A *simple k-module* is a graph composed of three tagged cycles, of lengths $3k$, $3k+1$, and $3k+2$ respectively. The base nodes of these tagged cycles are all joined in a cycle as illustrated in Figure 7.5. We say that the tag for the $3k$-cycle is the *primary tag* for the simple k-module.

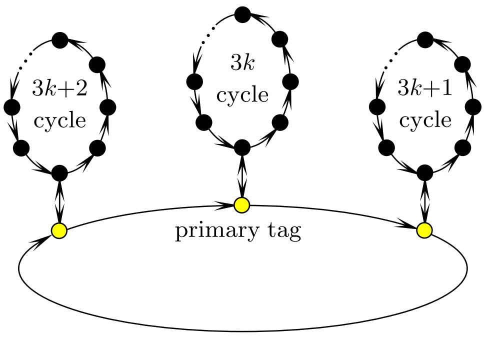

Figure 7.5: A simple k-module

Our witness that $\{y\}$ is not an isomorphism will be the primary tag of some simple k-module. We need only act if $\{y\}$ maps the primary tag of the simple k-module in $\mathfrak{B}_\sigma$ to the one in $\mathfrak{A}_n$. In that case, we extend the simple k-module to an augmented k-module.

Definition 7.3.2. An *augmented k-module* is a simple k-module where the primary tag is marked by a reflexive element. That is, if a is the primary tag, there is some c such that $(c\ R\ c)$ and $(c\ R\ a)$. Furthermore, every tag is connected to two cycles as illustrated in Figure 7.6.

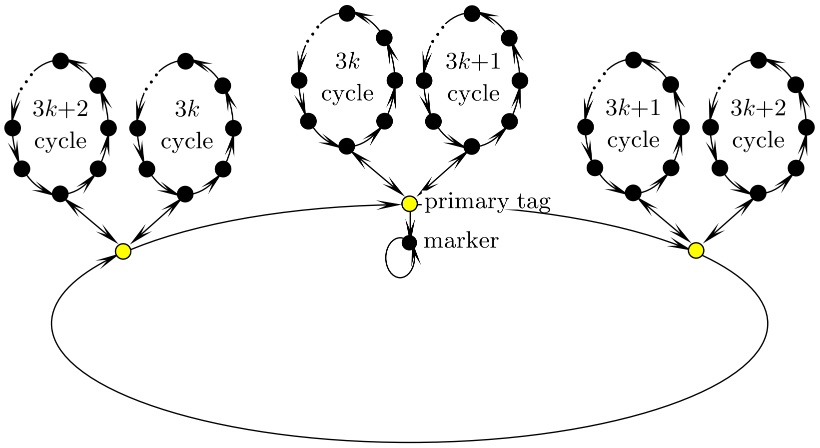

Figure 7.6: An augmented k-module

Note that there is more than one way to extend a simple k-module to an augmented k-module. In $\mathfrak{B}_\sigma$, we can mark the tag of the cycle of length $3k+1$, and then extend the k-module to the graph in Figure 7.6. In this case $\{y\}$ maps a tag without a marker to one with a marker, so it cannot be an isomorphism.

The problem with this approach is preserving positive requirement of higher priority. For any requirement $(P_{x,\tau})$ of higher priority, the candidate isomorphism $f_{x,\tau}$ for this requirement may already map the k-module in $\mathfrak{M}_x$ to the one in $\mathfrak{A}_n$. However, if the marker we add to the primary tag of $\mathfrak{A}_n$ later appears on the tag of the $3k+1$ cycle in $\mathfrak{M}_x$, $f_{x,\tau}$ can no longer be extended to an isomorphism.

For positive requirements properly to the left of our requirement $(N_{\sigma,y})$, this is not a problem. Every time we visit them, we will reset our k-module chosen to witness that $\{y\}$ is not an isomorphism, making sure that it is not in the domain of any of their candidate isomorphisms so far.

The real problem is for positive requirements directly above $(N_{\sigma,y})$ in the priority tree, because these do not get to reset our choice of k-module. Because of these positive requirements, we need an intermediate step in our diagonalization process.

Definition 7.3.3. A *marked k-module* is a simple k-module where the marker has been attached to the primary tag, as illustrated in Figure 7.7.

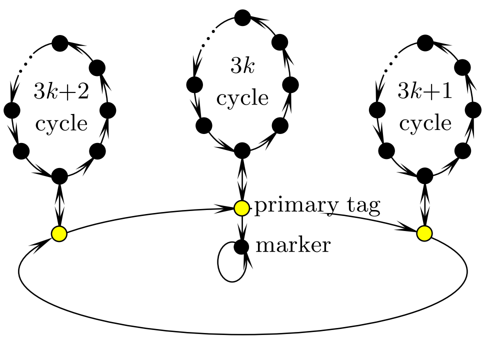

Figure 7.7: A marked k-module

Note that any simple k-module is uniquely extendable to a marked k-module, and any marked k-module is uniquely extendable to an augmented k-module. If $\{y\}$ maps the primary tag in $\mathfrak{B}_\sigma$ to the one in $\mathfrak{A}_n$, we mark the k-module in $\mathfrak{A}_n$, but do nothing to the k-module in $\mathfrak{B}_\sigma$. We then wait until the primary tags of all of the models for positive requirements of higher priority become marked. Once this happens, we can extend the k-modules in $\mathfrak{B}_\sigma$ and $\mathfrak{A}_n$ as before.

If this never happens, then our priority tree is such that we will never return to this requirement on the tree, and it will not be on our true path. Instead, some other model $\mathfrak{B}_\tau$ will witness that $\mathfrak{A}_n$ is not computably categorical, provided that $\mathcal{P}(n)$ does not hold.

Satisfying $(P_{x,\sigma})$ is much more straight-forward. For each k-module in $\mathfrak{A}_n$, we wait until a corresponding k-module appears in $\mathfrak{M}_x$ and extend our function $f_{x,\sigma}$ accordingly. The only problem is if a negative requirement of higher priority causes a k-module to be extended, possibly invalidating $f_{x,\sigma}$.

Again, negative requirements properly to the left are no problem; every time we visit them we reset $f_{x,\sigma}$ entirely. Our main concern is for negative requirements directly above $(P_{x,\sigma})$ in the priority tree, because these do not get to reset $f_{x,\sigma}$. There are only finitely many of these, so we simply do not ever define $f_{x,\sigma}$ on the k-modules being used by these requirements. Our true isomorphism from $\mathfrak{M}_x$ to $\mathfrak{A}_n$ will be a noneffective extension of $f_{x,\sigma}$ to these k-modules. As every k-module is finite, this is only finitely many elements, so our requirement will still be met.

7.4 The formal construction

As usual, we will proceed by stages. At each stage s, we will define a finite amount of our structure $\mathfrak{A}_n^s$, the witnesses $\mathfrak{B}_\sigma^s$ for the negative requirements, and the functions $f_{x,\sigma}^s$ for the positive requirements. Furthermore, we will need to make some guess as to the true path of our tree, so we will need some additional helper functions.

First and foremost, we need functions $z_s(\sigma)$, $w_s(\sigma)$. These functions will only be relevant for $\sigma \in \mathcal{T}$ of even length. They will approximate $\forall z\, \exists w\, \mathcal{Q}(x, y, z, w, n)$ for $\lceil \sigma \rceil = \langle x, y \rangle$.

Second, we need a function $m_s(\sigma)$. This function will only be relevant for σ corresponding to positive requirements. This function will denote the k-module, if any, for which this positive requirement is waiting for a marker to be attached to the primary tag.

Finally, we need a restraint function $r_s(\sigma)$. This function will only be relevant for σ of odd length; that is, $\sigma \in \mathcal{T}$ corresponding to the positive and negative requirements. For σ corresponding to positive requirements, $r_s(\sigma)$ identifies the k-modules which will not be part of domain of the function $f_{x,\rho}$ for this requirement. For σ corresponding to negative requirements, $r_s(\rho)$ identifies the k-module being used to satisfy this requirement.

For each $s \in \omega$, we let $\mathfrak{M}_e^s$ be the structure corresponding to the diagram of $\mathfrak{M}_e$ at stage s. In other words, $\mathfrak{M}_e^s$ is the graph whose edges and nodes are exactly those given by the atomic formulae

$$\operatorname{diag}_s(\mathfrak{M}_e) = \left\{ \xi \in \mathcal{L}^{\text{atm}}(\omega) \mid \{ e \}_s (\xi) \downarrow = 1 \right\} \tag{7.13}$$

For simplicity, we will slow down our enumeration of $\operatorname{diag}_s(\mathfrak{M}_e)$ enough so that $\mathfrak{M}_e^s \hookrightarrow \mathfrak{A}_n^s$ for all s. Note that this is indeed an enumeration of the structure $\mathfrak{M}_e$ only when $\mathfrak{M}_e \hookrightarrow \mathfrak{A}_n$. In particular,

$$\mathfrak{M}_e \not\hookrightarrow \mathfrak{A}_n \quad \Rightarrow \quad \exists s_0\, \forall s \geq s_0\, \mathfrak{M}_e^s = \mathfrak{M}_e^{s_0} \tag{7.14}$$

This will not prove a problem for our construction.

Given the above, for each $\sigma \in \mathcal{T}$, we define the σ-stages by induction on the length of σ.

(a) Every stage is a $\langle\rangle$-stage.

(b) If $s + 1$ is a σ-stage with $\lceil \sigma \rceil = \langle x, y \rangle$, we check if $\mathcal{Q}(x, y, z_s(\sigma), w_s(\sigma), n)$ holds. If so, $s + 1$ is a $\sigma^\frown\Pi$-stage; otherwise, it is a $\sigma^\frown\Sigma$-stage.

(c) If $s + 1$ is a σ-stage with $\lceil \sigma \rceil = P_{x,\rho}$, we check whether $\mathfrak{M}_x^{s+1}$ has expanded since the last σ-stage. Furthermore, if $m_s(\sigma) \neq -1$, we check whether the $m_s(\sigma)$-module of $\mathfrak{M}_x^{s+1}$ has a marker. If so, $s + 1$ is a $\sigma^\frown\infty$-stage; otherwise, it is a $\sigma^\frown f$-stage.

(d) If $s+1$ is a σ-stage with $\lceil \sigma \rceil = N_{\rho,y}$, we check whether the requirement $(N_{\rho,y})$ has been met. If so, $s + 1$ is a $\sigma^\frown m$-stage; otherwise it is a $\sigma^\frown u$-stage.

Finally, before we start our construction, we specify some invariants that will be met in our construction. We state them now, and will point out where they are all met in the construction. This will save us from having to go back and prove all of these invariants afterwards individually.

Lemma 7.4.1. *Our construction has the following properties.*

(a) For any s and $\mathfrak{B}_\sigma^s$, $\mathfrak{B}_\sigma^s \hookrightarrow \mathfrak{A}_n^s$.

(b) For all s and t, $r_s(\sigma) \neq r_t(\tau)$ whenever $\sigma \neq \tau$ and $r_s(\sigma) \neq -1$.

(c) If the k-modules in $\mathfrak{M}_e$ and $\mathfrak{A}_n$ are of the same type, and are fully enumerated in both models by stage s, then $f_{e,\sigma}^s$ can be extended to a isomorphism between the k-modules provided that $k > r_s(\sigma)$.

(d) Suppose that s is a σ_s-stage with $\lceil \sigma_s \rceil = N_{\rho,y}$ and $k = r_s(\sigma_s) \neq -1$. Then both $\mathfrak{A}_n^s$ and $\mathfrak{B}_\rho^s$ contain a k-module.

(e) A k-module of $\mathfrak{A}_n^s$ is augmented only if it satisfies some negative requirement $(N_{\sigma,y})$.

(f) Suppose that s is a σ_s-stage with $\lceil \sigma_s \rceil = N_{\rho,y}$. If $r_s(\sigma_s) = k$, then the k-module of $\mathfrak{B}_\rho^s$ is marked only if it is augmented.

(g) The function $m_s(\tau)$ can be reset at stage s only if s is a σ_s-stage for some $\tau <_L \sigma_s$.

(h) If $m_s(\sigma) \neq -1$, then $m_s(\sigma) = r_s(\rho)$ for some $\sigma^\frown\infty \prec \rho$.

(i) Suppose that s is a σ_s-stage with $\lceil \sigma_s \rceil = N_{\rho,y}$ and $k = r_s(\sigma_s)$. If the k-module of $\mathfrak{A}_n^{s-1}$ is marked, then it is marked in $\mathfrak{M}_x^s$ for all $\tau^\frown\infty \preceq \sigma$ with $\lceil \sigma \rceil = P_{x,\alpha}$.

(j) Suppose that s is a σ_s-stage with $\lceil \sigma_s \rceil = P_{x,\rho}$ and $k = m_s(\sigma_s) \neq -1$. Then the k-module is marked in $\mathfrak{A}_n^s$, but not in $\mathfrak{M}_x^s$.

<u>Stage $s = -1$:</u>
We let $\mathfrak{A}_n^{-1}$, $\mathfrak{B}_\sigma^{-1}$, and $f_{e,\sigma}^{-1}$ all be empty for the $\sigma \in \mathcal{T}$ for which they exist. Furthermore, we define $z_{-1}(\sigma) = w_{-1}(\sigma) = 0$ and $m_s(\sigma) = r_s(\sigma) = -1$ for all appropriate σ.

<u>Stage $s = t+1$:</u>
We enumerate a simple s-module into $\mathfrak{A}_n$. We let σ_s be the unique σ such that s is a σ-stage and $|\sigma| = s$. For all $\sigma_s <_L \tau$ with $\lceil \tau \rceil = P_{x,\rho}$, we reset $f_{x,\rho}$ and set $r_s(\tau) = -1$. Similarly, for all $\sigma_s <_L \tau$ with $\lceil \tau \rceil = N_{\rho,y}$, we set $r_s(\tau) = -1$. Furthermore, we reset $\mathfrak{B}_\rho$ if $y = 0$.

Now take any τ such that $\tau \preceq \sigma_s$. If $r_{s-1}(\tau) = -1$, we let $r_s(\tau)$ be some number k larger than any seen so far in our construction. In particular, $\mathfrak{A}_n^{s-1}$ does not have a k-module. This will ensure invariant (b) of Lemma 7.4.1. Furthermore, if $\lceil \tau \rceil = N_{\rho,y}$, we add a simple k-module to both $\mathfrak{A}_n$ and $\mathfrak{B}_\rho$, ensuring invariant (d).

If $r_{s-1}(\tau) \neq -1$, then we have three cases, depending on the nature of $\lceil \tau \rceil$.

Case $\lceil\tau\rceil = \langle x, y\rangle$: We define

(7.15)

$$z_s(\tau) = \begin{cases} z_{s-1}(\tau)+1 & \text{if } \tau\widehat{\ }\Pi \preceq \sigma_s \\ z_{s-1}(\tau) & \text{otherwise} \end{cases} \qquad w_s(\tau) = \begin{cases} w_{s-1}(\tau)+1 & \text{if } \tau\widehat{\ }\Sigma \preceq \sigma_s \\ 0 & \text{otherwise} \end{cases}$$

As all of the other parameters of our construction are not relevant to this node, we leave them untouched for τ.

Case $\lceil\tau\rceil = P_{x,\rho}$: If s is a $\tau\widehat{\ }f$-stage, there is nothing to do. So suppose s is a $\tau\widehat{\ }\infty$-stage. We let $r_s(\tau) = r_{s-1}(\tau)$ and look at any k-module in $\mathfrak{M}_x^s$ with $k \geq r_{s-1}(\tau)$ on which $f_{x,\tau}^s$ is not entirely defined. If the k-module in $\mathfrak{A}_n^s$ is not of the same type as in $\mathfrak{M}_x^s$, there is nothing to do. Otherwise, the invariant (c) says that we can extend $f_{x,\tau}^s$ so that it is an isomorphism between the two k-modules.

Finally, if $m_s(\tau) = \ell$ and the ℓ-module in $\mathfrak{M}_x^s$ is marked, we set $m_s(\tau) = -1$. Note that this ensures invariants (j) and (g)

Case $\lceil\tau\rceil = N_{\rho,y}$: If $y = 0$, we take every k-module in $\mathfrak{A}_n^s$ where k is not some $r_{s-1}(\alpha)$ for $\alpha \preceq \sigma_s$ with $\lceil\alpha\rceil = N_{\rho,i}$. These are all the k-modules of $\mathfrak{A}_n^s$ that have not yet been assigned to witness that $\{\, i \,\}$ is not a computable isomorphism from $\mathfrak{B}_\rho$ to $\mathfrak{A}_n$. We add each of these k-modules to $\mathfrak{B}_\rho$, if they are not already there, or only partially there. Furthermore, we do so in such a way to satisfy all requirements $(N_{\rho,z})$ for $z \leq s$ that we can. Note that ensures invariant (a).

If the requirement $N_{\rho,y}$ is met, there is nothing else to do. So suppose that $N_{\rho,y}$ is not met. We let $\ell = r_s(\tau) = r_{s-1}(\tau)$. If the ℓ-module in $\mathfrak{A}_n^s$ is simple, we check if $\{\, y \,\}$ maps the primary tags of the ℓ-modules to each other. If not, there is nothing to do. Otherwise, we mark this module in $\mathfrak{A}_n^s$ and set $m_s(\alpha) = \ell$ for all $\alpha\widehat{\ }\infty \prec \tau$. This will ensure invariants (f) and (h) above. If there is no such α, then we do not have to worry about preserving any isomorphisms, so we can augment the k-modules in both $\mathfrak{A}_n^s$ and $\mathfrak{B}_\rho^s$ to meet our requirement.

Note that invariant (j) ensures that for all $\alpha\widehat{\ }\infty \prec \tau$, $m_{s-1}(\alpha) = -1$. So we are not removing any previously defined restraints by assigning $m_s(\alpha) = k$. Furthermore, by our definition of a $\sigma\widehat{\ }\infty$-stage above, this assignment will also ensure invariant (i).

If the ℓ-module in $\mathfrak{A}_n^s$ is marked, then invariant (i) implies that we can augment the ℓ-module both in $\mathfrak{A}_n$ and $\mathfrak{B}_\rho$ to meet our requirement while still preserving the invariant (c). This will ensure the invariants (e) and (f).

Note that invariants (e) and (b) ensure that a k-module $\mathfrak{A}_n^s$ cannot be augmented if $N_{\rho,y}$ is not met. So this completes our construction.

Having finished our construction, we let $\mathfrak{A}_n = \bigcup_s \mathfrak{A}_n^s$. Furthermore, we let $\mathfrak{B}_\sigma = \bigcup_s \mathfrak{B}_\sigma^s$ and $f_{e,\sigma} = \lim_s f_{x,\sigma}^s$ whenever these limits exist.

As usual, we let TP denote the "true path" of our construction. That is, TP is the left most path in our priority tree visited infinitely often. First we need to prove some properties of our true path.

Lemma 7.4.2. *Let $\lceil \rho \rceil = \langle x, 0 \rangle$ be such that for all $y \in \omega$, there is some $\sigma \in \mathrm{TP}$ with $\lceil \sigma \rceil = N_{\rho,y}$. Then for all such σ, $\mathfrak{B}_\rho$ and $r(\sigma)$ are each reset finitely often, and requirement $(N_{\rho,y})$ is eventually met.*

Proof. By definition, $\lceil \sigma \rceil = N_{\rho,0}$ if and only if $\sigma = \rho^\frown\Sigma$. Therefore, as our hypotheses include $y = 0$, $\rho^\frown\Sigma \in \mathrm{TP}$. So as $\mathfrak{B}_\rho$ is only reset by requirements to the left of ρ, it is reset finitely often. Similarly, $r(\sigma)$ is only reset by requirements left of σ, so it is reset finitely often.

To see that $\mathfrak{B}_\rho \cong \mathfrak{A}_n$, take any k-module in $\mathfrak{A}_n$. By invariant (a), we need only show that $\mathfrak{B}_\rho$ has the same type of k-module. Suppose first that the k-module is not being used to satisfy some requirement $(N_{\rho,y})$. Then our construction adds or extends the k-module in $\mathfrak{B}_\rho$ the first stage we visit $\rho^\frown\Sigma$ after the k module has been added or extended in $\mathfrak{A}_n$. As $\rho^\frown\Sigma \in \mathrm{TP}$, the k-module in $\mathfrak{B}_\rho$ is the same type as in $\mathfrak{A}_n$.

Suppose now that $r_s(\sigma) = k$ for some stage s and $\lceil \sigma \rceil = N_{\rho,y}$. If $r_t(\sigma)$ is reset after stage s, then by invariant (b) this k-module is added or extended in $\mathfrak{B}_\rho$ at the next visit to $\rho^\frown\Sigma$. So suppose $\lim_t r_t(\sigma) = k$. Because of invariants (e) and (f), our only concern is that the k-module is marked in $\mathfrak{A}_n$ but simple in $\mathfrak{B}_\rho$. Let $t > s$ be the least stage such that the k-module becomes marked in $\mathfrak{A}_n$. Then it is clear from the construction that the k-module becomes augmented in both $\mathfrak{A}_n$ and $\mathfrak{B}_\rho$ at the next visit to $\rho^\frown\Sigma$. So again, the k-module in $\mathfrak{B}_\rho$ is the same type as in $\mathfrak{A}_n$.

Hence $\mathfrak{B}_\rho \cong \mathfrak{A}_n$. We need only show that there is no computable isomorphism $\{\, y \,\} : \mathfrak{B}_\rho \cong \mathfrak{A}_n$. Take any $y \in \omega$ and choose $\sigma \in \mathrm{TP}$ such that $\lceil \sigma \rceil = N_{\rho,y}$. We let s be some stage such that neither $\mathfrak{B}_\rho$ nor $r(\sigma)$ is reset after stage s and let $k = r_s(\sigma)$. If $\{\, y \,\}$ never maps the primary tags of the k-modules to each other, we are done. Otherwise, we let $t > s$ be the least stage at which we visit σ and we see that $\{\, y \,\}$ maps the primary tags to each other. Then the k-module is marked in $\mathfrak{A}_n$ at this stage, and our requirement is satisfied on the next visit to σ. $\square$

Lemma 7.4.3. *Suppose $\sigma \in \mathrm{TP}$ with $\lceil \sigma \rceil = P_{e,\rho}$ and $\mathfrak{M}_e \cong \mathfrak{A}_n$. Then $\sigma^\frown\infty \in \mathrm{TP}$.*

Proof. As $\sigma \in \mathrm{TP}$, there is some stage s_0 such that no node to the left of σ is visited after stage s_0. Take any stage $s \geq s_0$. We need to show that $\sigma^\frown\infty$ is visited some stage after s.

As $\mathfrak{M}_e \cong \mathfrak{A}_n$ and $\mathfrak{A}_n$ is infinite, there is some stage $t_0 \geq s$ such that $\mathfrak{M}_e^{t_0} \neq \mathfrak{M}_e^{t_0-1}$. As $\sigma \in \mathrm{TP}$, there is some $t \geq t_0$ such that t is σ-stage. This stage is also a $\sigma^\frown\infty$-stage whenever $m_t(\sigma) = -1$.

Suppose that $k = m_t(\sigma) \neq -1$. By invariant (j), this k-module is marked in $\mathfrak{A}_n^t$. By the invariants (e), (b), and (h), this k-module can only be augmented if we visit some node $\rho \succeq \sigma^\frown\infty$, and hence visit $\sigma^\frown\infty$. Similarly, by our choice of s_0 and the invariant (g), $m_t(\sigma)$ can only be reset if we visit $\sigma^\frown\infty$.

So suppose that $m(\sigma)$ is never reset. As $\mathfrak{M}_e \cong \mathfrak{A}_n$, there is some σ-stage $r \geq t$ at which the mark has appeared in $\mathfrak{M}_e$ and $\mathfrak{M}_e^r \neq \mathfrak{M}_e^t$. By construction, the least such stage is a $\sigma^\frown\infty$-stage. $\square$

Lemma 7.4.4. *Suppose that for all $x \in \omega$ there is some $\sigma \in \mathrm{TP}$ with $\lceil\sigma\rceil = P_{x,\rho}$. Then for all such x, $f_{x,\sigma}$ and $r(\sigma)$ are each reset finitely often, and requirement $(P_{x,\rho})$ is eventually met.*

Proof. The functions $f_{x,\sigma}$ and $r(\sigma)$ are each reset only by requirements to the left of σ, so clearly they are only reset finitely often. Take any $\sigma \in \mathrm{TP}$ with $\lceil\sigma\rceil = P_{x,\rho}$. Let s be a stage such that $f_{x,\sigma}$ and $r(\sigma)$ are never reset after s, with $m = \lim_s r_s(\sigma)$.

Suppose $\mathfrak{M}_x \cong \mathfrak{A}_n$. We need only show that $f_{x,\sigma}$ is a bijection between k-modules for $k > m$. Take any k-module in $\mathfrak{A}_n$ and let $t_0 > s$ be the stage after which it is never again extended, and at which it has entirely appeared in $\mathfrak{M}_x$. By Lemma 7.4.3, $\sigma^\frown\infty \in \mathrm{TP}$, so there is some stage $t \geq t_0$ at which we visit $\sigma^\frown\infty$. At this stage, $f_{x,\sigma}$ is extended to a bijection between the k-modules in these two structures, by invariant (c) of Lemma 7.4.1. □

We are now ready to prove Theorem 7.1.9. We need only show two things.

(a) If $\mathcal{P}(n)$ holds, we satisfy some positive requirement $(P_{x,\sigma})$ for each $x \in \omega$.

(b) If $\mathcal{P}(n)$ does not hold, there is some σ such that we satisfy the negative requirements $(N_{\sigma,y})$ for all $y \in \omega$.

Given the properties of the true path above, these are immediate from the following two lemmas.

Lemma 7.4.5. *If $\mathcal{P}(n)$ holds, then for all $x \in \omega$, there is some $\sigma \in \mathrm{TP}$ such that $\lceil\sigma\rceil = P_{x,\rho}$.*

Proof. Let x be least such that there is no $\sigma \in \mathrm{TP}$ with $\lceil\sigma\rceil = P_{x,\rho}$. As $\mathcal{P}(n)$ holds, there is some y such that $\forall z\, \exists x\, \mathcal{Q}(x, y, z, w, n)$. Let y be the least such. By Lemma 7.2.1, there is some $\tau \in \mathrm{TP}$ with $\lceil\tau\rceil = \langle x, y\rangle$. It is clear then from our construction in (7.15) that $\tau^\frown\Pi \in \mathrm{TP}$. But $\lceil\tau^\frown\Pi\rceil = P_{x,\tau}$. □

Lemma 7.4.6. *If $\mathcal{P}(n)$ does not hold, then there is some ρ such that for all $y \in \omega$, there is some $\sigma \in \mathrm{TP}$ with $\lceil\sigma\rceil = N_{\rho,y}$.*

Proof. As $\mathcal{P}(n)$ does not hold, there is some x such that $\forall y\, \exists z\, \forall w\, \neg\mathcal{Q}(x, y, z, w, n)$. Let x be the least such. If there is a greatest $\sigma \in \mathrm{TP}$ corresponding to a positive requirement, our claim is immediate from Lemma 7.2.1. Otherwise we know from Lemma 7.2.1 that there is some $\rho \in \mathrm{TP}$ with $\lceil\rho\rceil = \langle x, 0\rangle$. As $\exists z\, \forall w\, \neg\mathcal{Q}(x, 0, z, w, n)$, it is clear from (7.15) that there is some stage s after which every ρ-stage is a $\rho^\frown\Sigma$-stage. So $\rho^\frown\Sigma \in \mathrm{TP}$ with $\lceil\rho^\frown\Sigma\rceil = N_{\rho,0}$.

Let $\tau \in \mathrm{TP}$ be least element properly extending $\rho^\frown\Sigma$. It is clear from the description of the priority tree in Lemma 7.2.1 that $\lceil\tau\rceil = \langle x, 1\rangle$. So, continuing the above argument by induction on $y \in \omega$ achieves the desired result. □

7.5 Limitations of the construction

For all the work that Theorem 7.1.9 required, it is a rather anticlimactic result. While it is a lower bound for the complexity of computable categoricity, it is a rather weak in comparison to the upper bound. For all we know, computable categoricity, like hyperarithmetic categoricity, could be Π^1_1-complete.

However, there is an important difference between these two properties that shows why it is so difficult to construct a lower bound for computable categoricity. The primary difficulty is with the requirements (Π^0_4) in section 7.2. When we construct either a hyperarithmetically or computably categorical structure $\mathfrak{A}$, we need to witness an isomorphism $f : \mathfrak{A} \to \mathfrak{M}_e$ of the appropriate complexity whenever $\mathfrak{A} \cong \mathfrak{M}_e$. In our construction above, this isomorphism f is built as a limit of finite partial functions f_s from $\mathfrak{A}$ to $\mathfrak{M}_e$. This requires us to choose, at each step, f_s so that it extends to an isomorphism between $\mathfrak{A}$ and $\mathfrak{M}_e$. Therefore, in this type of construction, f will be computable in the predicate

$$\mathrm{EXT}_{\mathfrak{A}}(\sigma, e) =_{\mathrm{def}} \exists f \left((f : \mathfrak{A} \cong \mathfrak{M}_e) \wedge \forall i < |\sigma| \, (\sigma(i) = f(i)) \right) \tag{7.16}$$

For hyperarithmetic categoricity, this is not a problem. Since f only needs to be hyperarithmetic, we just need to construct our structures $\mathfrak{A}_n$ so that $\mathrm{EXT}_{\mathfrak{A}_n}$ is hyperarithmetic. In particular, we used the fact that EXT is hyperarithmetic for the computable ordinals in our proof of Theorem 5.5.3.

However, for computable categoricity, we need $\mathrm{EXT}_{\mathfrak{A}}$ to be computable. Note that

$$\mathrm{ISOM}_{\mathfrak{A}}(e) \quad \Leftrightarrow \quad \mathrm{EXT}_{\mathfrak{A}}(\langle\rangle, e) \tag{7.17}$$

So $\mathrm{ISOM}_{\mathfrak{A}}$ is computable, whenever $\mathrm{EXT}_{\mathfrak{A}}$ is. Therefore

(7.18)

$$\mathrm{CAT}_c(e) \Leftrightarrow \forall i \left[\mathrm{ISOM}_{\mathfrak{M}_e}(i) \to \exists j \begin{pmatrix} \forall n(\{j\}(n)\downarrow) \wedge \forall m \, \exists n \left(\{j\}(n) = m\right) \wedge \\ \forall n, m \left(\{j\}(n) = \{j\}(m) \to n = m\right) \wedge \\ \forall^{\mathrm{atm}} \xi, \overline{c} \left(\mathfrak{M}_e \models \xi(\overline{c}) \leftrightarrow \mathfrak{M}_i \models \xi(\{j\}(\overline{c}))\right) \end{pmatrix} \right]$$

where $\forall^{\mathrm{atm}}\xi$ means that ξ ranges over all atomic formulae. So it is clear from (7.18) that computable categoricity is Π^0_4 when it is restricted to structures $\mathfrak{A}$ for which $\mathrm{EXT}_{\mathfrak{A}}$ is computable. Therefore, Theorem 7.1.9 is the best that we can do with this method of constructing our isomorphisms.

Bibliography

[1] C. Ash, *Recursive labelling systems and stability of recursive structures in hyperarithmetical degrees*, Trans. Amer. Math. Soc. **298** (1986), no. 2, 497–514.

[2] C. Ash and J. Knight, *Computable structures and the hyperarithmetical hierarchy*, North Holland, Amsterdam, to appear.

[3] C. J. Ash and A. Nerode, *Intrisically recursive relations*, Aspects of Effective Algebra (J. N. Crossley, ed.), U.D.A. Book Co., Steel's Creek, Australia, 1981.

[4] J. Baldwin and D. Kueker, *Algebraically prime models*, Ann. Math. Logic **20** (1981), 289–330.

[5] K. J. Barwise and A. Robinson, *Completing theories by forcing*, Ann. Math. Logic **2** (1970), 119–142.

[6] W. Baur, *Uber rekursive strukturen*, Inventiones math. **23** (1974), 89–95 (German).

[7] P. Bernays and M. Schönfinkel, *Zum entscheidungsproblem der mathematischen logik*, Math. Annalen **99** (1928), 342–372 (German).

[8] A. Chistov and D. Grigor'ev, *Complexity of quantifier elimination in the theory of algebraically closed fields*, Mathematical Foundations of Computer Science (M. Chytil and V. Koubek, eds.), Lecture Notes in Comp. Sci., no. 176, Springer-Verlag, Berlin, 1984, pp. 17–31.

[9] P. J. Cohen, *The independence of the continuum hypothesis*, I. Proc. Nat. Acad. Sci. USA **50** (1963), 1143–1148.

[10] A. Ehrenfeucht, *Applications of games to the completeness problem for formalized theories*, Fundamenta Math. **48** (1961), 129–141.

[11] Y. Ershov and S. Goncharov, *Elementary theories and their constructive models*, Handbook of Recursive Mathematics (Y. Ershov, S. Goncharov, A. Nerode, and J. Remmel, eds.), Studies in Logic and the Foundations of Mathematics, vol. 138, North Holland, Amsterdam, 1999, pp. 115–166.

[12] R. Fraïssé, *Applications de γ-opérateurs au calcul logique du premier éschelon*, Z. Math. Logik Grundlag. Math. **2** (1961), 76–92.

[13] H. Friedman and L. Stanley, *A Borel reducibility theory for classes of countable structures*, J. Symbolic Logic **54** (1989), 894–914.

[14] K. Gödel, *Die Vollständigkeit der Axiome des logischen Funktionenkalküls*, Monashefte für Math. u. Phys. **37** (1930), 349–360.

[15] ______, *Über formal unentscheidbare Sätze der Principia Mathemtaica und verwandter System I*, Monashefte für Math. u. Phys. **38** (1931), 173–198.

[16] S. Goncharov, *Autostability and computable families of constructivations*, Algebra and Logic **14** (1975), no. 6, 392–409.

[17] S. Goncharov and A. Nurtazin, *Constructive models of complete solvable theories*, Algebra and Logic **12** (1973), no. 2, 67–77.

[18] T. Grilliot, *Omitting types: application ot recursion theory*, J. Symbolic Logic **37** (1972), 81–89.

[19] V. Harizanov, *Pure computable model theory*, Handbook of Recursive Mathematics (Y. Ershov, S. Goncharov, A. Nerode, and J. Remmel, eds.), Studies in Logic and the Foundations of Mathematics, vol. 138, North Holland, Amsterdam, 1999, pp. 3–114.

[20] L. Harrington, *Recursively presentable prime models*, J. Symbolic Logic **39** (1974), 305–309.

[21] J. Harrison, *Recursive pseudowellorderings*, Trans. Amer. Math. Soc. **131** (1960), 526–543.

[22] J. Heintz, *Definability and fast qauntifier elimination in algebraically closed fields*, Theoretical Comp. Sci. **24** (1983), 239–277.

[23] L. Henkin, *The completeness of the first-order functional calculus*, J. Symbolic Logic **14** (1949), 159–166.

[24] D. Hirschfeldt, *Degree spectra of computable structures*, Ph.D. thesis, Cornell University, 1999.

[25] W. Hodges, *Building models by games*, London Math. Society Student Texts, vol. 2, Cambridge University Press, London, 1985.

[26] ______, *Model theory*, Encyclopedia of Math. vol. 42, Cambridge University Press, London, 1993.

[27] B. Jónnson, *Universal relational systems*, Math. Scand. **4** (1956), 193–208.

[28] ______, *Homogeneous universal relational systems*, Math. Scand. **8** (1960), 137–142.

[29] A. Kechris and A. Louveau, *Descriptive set theory and the structure of sets of uniqueness*, London Math. Society Lecture Note Series, vol. 2, Cambridge University Press, London, 1987.

[30] H. J. Keisler, *Forcing and the omitting types theorem*, Studies in Model Theory (M. D. Morley, ed.), MAA Studies in Mathematics, vol. 8, Mathematical Association of America, Buffalo, NY, 1973, pp. 96–133.

[31] B. Khoussainov and R. Shore, *Computable isomorphisms, degree spectra of relations, and scott families*, Ann. Pure and Applied Logic **93** (1998), 153–193.

[32] S. Kleene, *Arithmetical predicates and function quantifiers*, Trans. Amer. Math. Soc. **79** (1955), 312–340.

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

[34] P. Lindström, *On model-completeness*, Theoria **30** (1964), 183–196.

[35] T. Millar, *Omitting types, type spectrums, and decidability*, J. Symbolic Logic **48** (1983), 171–181.

[36] T. S. Millar, *Foundations of recursive model theory*, Ann. Math. Logic **13** (1978), 305–320.

[37] M. Morley and R. Vaught, *Homogeneous universal models*, Math. Scand. **11** (1962), 37–57.

[38] M. Peretyat'kin, *Finitely axiomatizable theories*, Siberian School of Algebra and Logic, New York, 1997.

[39] M. Pouzet, *Modèle universel d'une théorie n-complète*, C.R. Acad. Sci. Paris Sér. A-B **274** (1972), A433–A436.

[40] M. O. Rabin, *Nonstandard models and independence of the induction axiom*, Essays on the Foundations of Mathematics (Y. Bar-Hillel, ed.), North Holland, Amsterdam, 1962, pp. 287–299.

[41] A. Robinson, *Complete theories*, North Holland, Amsterdam, 1956.

[42] H. Rogers, *Theory of recursive functions and effective computability*, McGraw-Hill, New York, 1967.

[43] G. Sacks, *Higher recursion theory*, Springer-Verlag, Berlin, 1990.

[44] D. Scott, *Logic with denumerably long formulas and finite strings of quantifiers*, The Theory of Models (J. Addison, L. Henkin, and A. Tarski, eds.), Proc. 1963 Internat. Symposium at Berkeley, North Holland, Amsterdam, 1965, pp. 329–82.

[45] S. Shelah, *Classification theory and the number of non-isomorphic models*, second ed., North Holland, Amsterdam, 1978.

[46] R. Soare, *Recursively enumerable sets and degrees*, Springer-Verlag, Berlin, 1987.

[47] E. Steinitz, *Algebraische theorie der körper*, Reine Angew. Math. **137** (1910), 167–309 (German).

[48] R. Vaught, *Denumerable models of complete theories*, Infinitistic methods, Proc. Symp. Foundations of Math. Warsaw 1959 (Warsaw), Panstwowe Wydawnictwo Naukowe, 1961, pp. 303–321.