DECIDABILITY IN THE HYPERDEGREES AND A THEOREM OF HYPERARITHMETIC ANALYSIS

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

James Barnes

August 2018

DECIDABILITY IN THE HYPERDEGREES AND A THEOREM OF HYPERARITHMETIC ANALYSIS

James Barnes, Ph.D.

Cornell University 2018

In this thesis we explore two different topics: the complexity of the theory of the hyperdegrees, and the reverse mathematics of a result in graph theory.

For the first, we show the Σ_2 theory of the hyperdegrees as an upper-semilattice is decidable, as is the Σ_2 theory of the hyperdegrees below Kleene's $\mathcal{O}$ as an upper-semilattice with greatest element. These results are related to questions of extensions of embeddings into both structures, i.e., when do embeddings of a structure extend to embeddings of a superstructure.

The second part is joint work with Richard Shore and Jun Le Goh. We investigate a theorem of graph theory and find that one formalization is a theorem of hyperarithmetic analysis: the second such example found, as it were, in the wild. This work is ongoing, and more may appear in future publications.

BIOGRAPHICAL SKETCH

James Barnes was born and raised in the UK. In his early years he loved mathematics, but his relationship with it became more mixed when he was a teen. Out of worry he would be bored, he decided to pursue a bachelor's degree in mathematics and philosophy at the Warwick University rather than just studying mathematics.

He found the undergraduate philosophy curriculum mostly unengaging (perhaps because he was too lazy to read), but was drawn to the technical aspects: logic. As has usually been the case previously, he managed to get by with heavy cramming before exams and decided that he wanted to learn more logic.

At the suggestion of his undergraduate advisors Walter Dean and Adam Epstein, he applied to graduate schools in the United States and was accepted into Cornell University. On arrival, he quickly realized that he was not interested in a life of research, and instead was more interested in teaching. His graduate advisor Richard Shore was kind enough to support him in this endeavor and has cajoled him into completing sufficient research to conclude his PhD.

To my great-grandmother Stella Bland who passed while I was making this.

ACKNOWLEDGEMENTS

A full accounting of the people who helped me and the ways I was supported would be, as it were, beyond the scope of this document. As such I will try to keep this to a few key players.

Firstly, I should thank the mathematics teachers I had when I was an unhappy teen: David Ellis and Chris Harrison. I was a tiresome person at that point in my life. Secondly, my academic advisors at Warwick need a mention for encouraging me to continue my education after a first round of university. Thank you both Adam Epstein and, in particular, Walter Dean for giving me a model of the kind of person I wanted to be.

Thirdly, I would like to thank my committee Richard Shore, Dexter Kozen, and Anil Nerode for their time and patience. In particular, my chair Richard Shore whose was understanding of my limitations as a researcher.

On the non-academic side, I want to thank the staff in the mathematics department especially Melissa Totman with whom I spent many procrastinatory hours.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Recursion theory is a branch of mathematics that began in the 1930s. Various mathematicians were trying to give abstract definitions to formalize the notion of a computable function. Many different models were proposed and they all turned out to define the same class of functions.

One of the most attractive models is that of a **Turing machine**. A Turing machine is a kind of abstract computer; the machine is sitting on a tape that has been divided into cells. The machine has a head that can read what is printed on the cell it is above, writing apparatus to delete what is written and write something new, and can move along the tape. The machine also has an internal state that tells it what action to take given what it is reading, and what state to transition into after taking that action.

We say a partial function $f : \omega^n \to \omega$ is **Turing computable** (or **recursive**, or **computable**) if there is a Turing machine T so that for each $(x_1, \ldots, x_n) \in \mathrm{dom}(f)$ if you start the Turing machine with $(x_1, \ldots, x_n)$ on the tape, then it will eventually halt with $f(x_1, \ldots, x_n)$ (and nothing else) on the tape. The machine is allowed to run for as long as it likes, and to use as much tape as it likes. We call a set Turing computable (or recursive, or computable) if its characteristic function is computable.

It is clear that every Turing computable function is computable in the intuitive sense: you could actually sit there and compute it given enough time and paper (or rather, this would be clear if I provided a more precise definition of a Turing machine). The Church-Turing thesis states that the converse is true:

Church-Turing Thesis. *A partial function $f : \omega^n \to \omega$ is intuitively computable if and only if it is Turing computable.*

The thesis is not amenable to proof, because there is no formal definition of "intuitively computable". It is best thought of as a conceptual analysis, or, perhaps, as a definition of computable.

In his PhD thesis [26] Turing extended his notion of Turing machine to that of a Turing machine with oracle. The idea of a Turing machine with oracle is that the Turing machine is given access to answers to questions of the form "Is $x \in X$" for some $X \subseteq \omega$ and to branch depending on the answer it gets. By fixing the Turing machine but varying the oracle the action of the Turing machine changes. For instance, if you give the Turing machine an oracle for a set such that a Turing machine without an oracle cannot determine membership, then the augmented machine can do something that the simpler one could not.

A Turing machine with oracle also allows us to try to compare different subsets of ω to determine whether one is more computationally complex than another. If there is a Turing machine with oracle X that computes the characteristic function of Y, then we write $Y \leq_T X$ (we read this "Y is Turing reducible to X"). In this case, we think of X as containing more information than Y because if only we could determine membership in X, then we could compute membership in Y.

1.1 The Turing degrees and other preliminaries

We give a brief overview of some basics in recursion theory that will be needed later. For more details consult Lerman [15], Rogers [18], or Soare [24].

The relation $\leq_T$ of Turing reducibility was initially studied in Kleene's and Post's classic *The Upper Semi-Lattice of Degrees of Recursive Unsolvability* [13]. The first facts to observe are that $\leq_T$ is reflexive and transitive. It is reflexive because there is a Turing machine which asks "Is $x \in X$" and prints "yes" if it is returned a yes and "no" if it is returned a no. It is transitive because if there is a program to compute X from Y and a program to compute Y from Z, then we can compute X from Z by running the first program, but replacing any of the points in the program where we ask "is $y \in Y$" with the program that computes Y from Z.

In technical terms, $\leq_T$ is a preorder. In Appendix A the reader can find the technical definitions and some algebraic results involving orders, lattices, and semi-lattices that are used in the main body of the text. As observed there, given a preorder one can define an equivalence relation. For $\leq_T$ we have $\equiv_T$ defined by:

$$X \equiv_T Y \iff X \leq_T Y \text{ and } Y \leq_T X.$$

This equivalence relation introduces a quotient on the powerset of ω, which we denote $\mathcal{D}_T$. The relation $\leq_T$ descends to the equivalence classes comprising $\mathcal{D}_T$ and endow it with the structure of a partially ordered set (poset). We call the equivalence class of X its **Turing degree**. Turing degrees are usually denoted by bold roman lowercase letters, and members of them by the uppercase version, e.g., the degree of X is $\mathbf{x}$. We often confuse a set and its degree.

The degree of the emptyset is the minimum element in $\mathcal{D}_T$. There is also an operator $\oplus$ on subsets of ω defined by

$$X \oplus Y = \{2x : x \in X\} \cup \{2y + 1 : y \in Y\}.$$

It can be shown that $\oplus$ is well-defined on Turing degrees, i.e, if $X_1 \equiv_T X_2$ and $Y_1 \equiv_T Y_2$, then $X_1 \oplus Y_1 \equiv_T X_2 \oplus Y_2$. Furthermore, the degree of $X \oplus Y$ is the least

upper bound of X and Y, therefore, $\mathcal{D}_T$ is an upper-semilattice with least element. There is also an infinite version of this operation: Let $\langle \vec{x} \rangle$ be a computable encoding of strings such as $\langle x_1, \ldots, x_n \rangle = \prod_{i=1}^{n} p_i^{x_i+1}$ where p_i is the ith prime. Then for each infinite sequence $\{X_n\}_{n=0}^{\infty}$ of sets

$$\bigoplus_{n=0}^{\infty} X_n = \{\langle x, n \rangle : x \in X_n\}.$$

Given X we define $X^{[n]} = \{x : \langle x, n \rangle \in X\}$, the nth **column** of X. The operator $\bigoplus$ is not well defined on degrees, however, X_n is Turing reducible to $\bigoplus X_n$ for each n.

Let Φ_e for $e \in \omega$ denote the standard enumeration of Turing machines with oracles, so $\Phi_0^X, \Phi_1^X, \Phi_2^X, \ldots$ enumerates the partial functions that are recursive in X. With this enumeration we can define the **(Turing) jump** of X to be

$$X' = \{e : \Phi_e(e)^X \downarrow\}.$$

($\downarrow$ is an abbreviation for halts). It can be shown that X' is strictly greater than X in Turing degree. We define $X^{(0)} = X$ and $X^{(n+1)} = (X^{(n)})'$.

A set Y is called **recursively enumerable** in X (X-r.e.) if Y is the domain of some X partial computable function. We denote the eth X-r.e. set by $W_e^X = \mathrm{dom}(\Phi_e^X)$. Equivalently Y is r.e. in X if it is the range of some partial X-computable function, or if Y is empty or is the range of some total X-computable function. The jump X' is r.e. in X for every X, indeed, it is the most complicated X-r.e. set in the sense that $X' \geq_T Y$ for every Y that is r.e. in X.

1.1.1 Definability and the arithmetic degrees

We say that a set $X \subseteq \omega$ is **definable in (first-order) arithmetic** if there is a formula $\varphi(x)$ in the first order language with $\{<, +, \times, 0, 1\}$ such that

$$n \in X \iff \omega \vDash \varphi(\overbrace{1 + 1 + \cdots + 1}^{n \text{ times}}).$$

We are particularly interested in formulas in prenex normal form i.e. formulas with all quantifiers at the front. If φ is a formula and t is a term (not containing x), then we let $(\forall x < t)\varphi$ and $(\exists x < t)\varphi$ abbreviate $(\forall x)[x < t \to \varphi]$ and $(\exists x)[x < t \wedge \varphi]$, respectively. We call instances of $(\exists x < t)$ and $(\forall x < t)$ **bounded quantifiers**. A formula φ is said to be Σ^0_0 and Π^0_0 if all quantifiers occurring are bounded. We call a formula Σ^0_{n+1} if it is in the form $(\exists x_1) \cdots (\exists x_m)\psi$ where ψ is Π^0_n, and similarly call a formula Π^0_{n+1} if it is in the form $(\forall x_1) \cdots (\forall x_m)\psi$ where ψ is Σ^0_n. A set is Σ^0_n if it is defined by a Σ^0_n formula, and similarly for Π^0_n. A set is Δ^0_n if it is both Σ^0_n and Π^0_n.

This syntactic characterization of sets is turns out to have close ties with computability: X is computable iff it is Δ^0_1, and X is r.e. iff it is Σ^0_1. If you add a set parameter symbol `Y` to our language and interpret it as a set Y, then $X \leq_T Y$ iff X is Δ^0_1 in this extended language (we write $\Delta^0_1(Y)$), and similarly for r.e. in Y (and $\Sigma^0_1(Y)$).

The collection of sets that are definable in first-order arithmetic are called the **arithmetic sets**. We say that X is **arithmetic in** Y if X is definable in first-order arithmetic with a predicate for membership in Y. Equivalently, X is arithmetic in Y if X is Turing reducible to $Y^{(n)}$ for some n. This relation (written $\leq_a$) is a second notion of reducibility from one set to another; it is also transitive and reflexive and induces a quotient structure $\mathcal{D}_a$ in the same way that $\leq_T$ does. It has

a least degree: the degree of the arithmetic sets. Furthermore, the join operator $\oplus$ defined above is also a join for this partial order. There is even a notion of jump called the **ω-jump**

$$X^{(\omega)} = \bigoplus_{n \in \omega} X^{(n)}.$$

1.1.2 Other reducibilities

We can be even more permissive in what is counted as a reduction. At the most extreme is the notion of X being **constructible** in Y; X is constructible in Y if X is in $L(Y)$ where L is the constructible universe of sets (in the set theoretic sense). On the very restrictive end, we can allow fewer reductions. In computer science there is a whole zoo of these notions which require the algorithms to have some asymptotic bound on the time the algorithm is allowed to run (or the space it is allowed to use as memory).

One notion that will be mentioned later is **many-one reducibility**. X is many-one reducible to Y (written $X \leq_m Y$) if there is a recursive function f such that $x \in X$ iff $f(x) \in Y$. One way to think about this is as a Turing reduction, but for each x you are only allowed to ask one question about Y. If the f happens to be injective, then it is called a 1-reduction.

However, the major focus of this document is on hyperarithmetic reducibility. This reducibility is intermediate between arithmetic and constructible. There is one way to define this reducibility that is very simple to understand: X is **hyperarithmetic** in Y if X is Δ^1_1 definable in second-order arithmetic with a predicate for membership in Y, i.e., there are formulas φ, ψ in second-order arithmetic with a predicate for Y with only existential second-order quantifiers or only universal

second-order quantifiers, respectively, that define membership in X. We will see, later, that this definition is transitive (it is obviously reflexive).

Unfortunately, this definition does not make it clear how we would go about proving anything about the reducibility. It turns out that there is an equivalent definition of hyperarithmetic reducibility that is more "bottom-up". In this case, X is hyperarithmetic in Y if X is Turing reducible to the αth jump of Y for some ordinal α with a Y-computable representation. Much care is needed to make this definition precise. Much of the rest of the next chapter is devoted to the various definitions and theorems required.

One further note is that the definitions above are given in terms of reducibility of sets. There are almost identical definitions for reducibilities between functions $f : \omega \to \omega$. By identifying a function with its graph and using a paring function, you can regard a function as a subset of ω, and you can regard a subset of ω as a function by identifying it with its characteristic function.

CHAPTER 2

THE FUNDAMENTALS OF HYPERARITHMETIC THEORY

Here we define the hyperarithmetic sets, hyperarithmetic reduction, the hyperjump, and various other pieces needed for our later work. None of the theorems in this chapter are mine; they all occur in Sacks' *Higher Recursion Theory* [19]. Where possible, I give the original citation for each result, as well as a reference within Sacks' text.

2.1 Kleene's notations for ordinals

A predicate $R \subseteq \omega^\omega \times \omega$ is **recursive** if there an index e such that for all $f \in \omega^\omega$ and $n \in \omega$, $\Phi_e^f(n)$ converges and $\Phi_e^f(n) = 0$ iff $R(f, n)$ is true. It is routine to extend this notion to a predicate $R \subseteq (\omega^\omega)^n \times \omega^m$ that takes multiple functions or numbers. Additionally, our predicate could be a subset of just ω^ω, or 2^ω, or just ω and so on.

A predicate is **analytical** if it is built up from recursive predicates by propositional connectives, natural number quantifiers, and function quantifiers. A predicate is **arithmetic** if it is analytical but does not use any function quantifiers. Kleene [12] showed that the analytical predicates can be put into a standard form with an initial (potentially empty) block of alternating function quantifiers and then an arithmetical predicate.

Definition 2.1 (Analytical hierarchy)**.** An analytical predicate is Σ^1_0 and Π^1_0 if it is arithmetic. Then, inductively, a analytical predicate is Σ^1_{n+1} if it is equivalent to a predicate $(\exists f)R$ where R is Π^1_n, and it is Π^1_{n+1} if its complement is Σ^1_{n+1}. A predicate is Δ^1_n if it is both Σ^1_n and Π^1_n.

We relativise this definition to a set X (and write $\Sigma^1_n(X), \Pi^1_n(X)$, and $\Delta^1_n(X)$) by defining $\Sigma^1_0(X)$ and $\Pi^1_0(X)$ predicates to be those with a definition that is arithmetic over X, then the inductive step is as in the unrelativised case.

By Kleene's normal form theorem, a predicate is Σ^1_n or Π^1_n, respectively, iff it is equivalent to a predicate of the form

$$(\exists f_1)(\forall f_2)\cdots(Pf_n)R, \quad \text{or} \quad (\forall f_1)(\exists f_2)\cdots(Qf_n)S, \quad \text{respectively,}$$

where R and S are arithmetic, and P is $\exists$ is n is odd, and if it is $\forall$ if n is even, and Q is the dual of P. Kleene's normal form theorem still holds when we relativise.

It turns out that we are particularly interested in the Δ^1_1 and Π^1_1 predicates. Here we state a technical result of Spector's that shows a set is Π^1_1 which has a number of applications and another technical result that applies to functions in particular.

Theorem 2.2 (Spector [25]; I.1.6 in Sacks)**.** *Suppose $A(X)$ is a Σ^1_1 predicate of sets (i.e., $A \subseteq 2^\omega$). Then $\{n : n \in X$ for each X satisfying $A\}$ is Π^1_1, and if there is a unique X satisfying A, then that X is Δ^1_1.*

Proposition 2.3 (I.1.7 in Sacks)**.** *A function f is Σ^1_n iff it is Π^1_n.*

As a first application of Theorem 2.2 we define the set $\mathcal{O}$. The most important Π^1_1 set.

Definition 2.4 (Kleene's $\mathcal{O}$)**.** We define an arithmetical predicate $A(X)$ of sets by four clauses:

(1) $\langle 1, 2\rangle \in X$;

(2) $(\forall x, y)[\langle x, y\rangle \in X \to \langle y, 2^y\rangle \in X]$;

(3) $(\forall x, y, z)[\langle x, y\rangle \in X \wedge \langle y, z\rangle \in X \to \langle x, z\rangle \in X]$;

(4) $(\forall e)[\Phi_e \text{ is total} \wedge (\forall x)[\langle \Phi_e(x), \Phi_e(x+1)\rangle \in X] \to 3 \cdot 5^e \in X]$.

Let $<_{\mathcal{O}}$ be the set $\{\langle a, b\rangle : \langle a, b\rangle \in X \text{ for all } X \text{ satisfying } A\}$. Let $\mathcal{O} = \{a : (\exists b)[a <_{\mathcal{O}} b \vee b <_{\mathcal{O}} a]\}$.

We think of $\mathcal{O}$ as a collection of names for ordinals: 1 represents the ordinal 0; if a represents an ordinal, then 2^a represents its successor; and if Φ_e enumerates a strictly increasing list of names for ordinals, then $3 \cdot 5^e$ represents the limit of the ordinals represented. The binary relation $<_{\mathcal{O}}$ attempts to compare the representations of ordinals, however, it is not total because each limit ordinal can be assigned many different names (all of the form $3 \cdot 5^e$) and we do not try to compare these under $<_{\mathcal{O}}$. We formalize this with the following proposition that allows us to define a map from $\mathcal{O}$ to the ordinals that maps names to what they represent.

Proposition 2.5 (I.2.2 in Sacks)**.** *The sets $<_{\mathcal{O}}$ and $\mathcal{O}$ are Π^1_1; $<_{\mathcal{O}}$ is a wellfounded partial order with least element 1; and if $y \in \mathcal{O}$, then $\{x : x <_{\mathcal{O}} y\}$ is linear.*

The proof of the first part of this proposition is to observe that $<_{\mathcal{O}}$ is the intersection of all solutions to an arithmetical predicate, and hence Π^1_1 by Theorem 2.2. $\mathcal{O}$ is defined in an arithmetic way from $<_{\mathcal{O}}$ and, hence, is Π^1_1 too. The fact that $<_{\mathcal{O}}$ is wellfounded allows us to define objects on $\mathcal{O}$ by induction. One of the key ones is given next:

Definition 2.6 ($|\cdot|$)**.** We define a function $|\cdot|$ from $\mathcal{O}$ to the ordinals by induction

on $<_{\mathcal{O}}$:

$$|a| = \begin{cases} 0 & \text{if } a = 1; \\ |b| + 1 & \text{if } a = 2^b; \\ \lim\limits_{n\to\infty} |\Phi_e(n)| & \text{if } a = 3 \cdot 5^e. \end{cases}$$

If $\alpha \in \text{range}(|\cdot|)$, then we call α **constructive**, and an a such that $|a| = \alpha$ is a **notation** for a.

It can be shown that the constructive ordinals form a proper initial segment of ω_1: properness follows from the fact that $\mathcal{O}$ is countable and ω_1 has cofinality ω_1, which is not countable. We show that the constructive ordinals are an initial segment by induction; our hypothesis is that if α is constructive, then every $\beta \leq \alpha$ is constructive. The case $\alpha = 0$ is obvious. For $\alpha+1$ suppose $\alpha+1$ is constructive, then

$$\begin{aligned} \{\beta \leq \alpha + 1 : \beta \in \text{range}\,|\cdot|\} &= \{\alpha + 1\} \cup \{\beta \leq \alpha : \beta \in \text{range}\,|\cdot|\} \\ &= \{\alpha + 1\} \cup (\alpha + 1) && \text{(by induction)} \\ &= \alpha + 2. \end{aligned}$$

Consequently, $\alpha+1$ satisfies the inductive hypothesis. Finally if α is a constructive limit ordinal, then $\alpha = |3 \cdot 5^e|$ for some e. By the induction hypothesis, for each n, $\alpha_n = |\Phi_e(n)|$ satisfies the induction hypothesis. Consequently

$$\begin{aligned} \{\beta \leq \alpha : \beta \in \text{range}\,|\cdot|\} &\supseteq \bigcup_{n\in\omega} \{\beta \leq \alpha_n : \beta \in \text{range}\,|\cdot|\} \\ &\supseteq \bigcup_{n\in\omega} (\alpha_n + 1) && \text{(by induction)} \\ &= \alpha && \text{as } \alpha = \lim_{n\to\infty} \alpha_n \end{aligned}$$

As α is constructive by hypothesis,

$$\{\beta \leq \alpha : \beta \in \text{range}(|\cdot|)\} \supseteq \alpha \cup \{\alpha\} \supseteq \alpha + 1.$$

Which concludes the limit case.

Definition 2.7 (ω_1^{CK})**.** We denote the least nonconstructive ordinal by ω_1^{CK}.

We know that $\mathcal{O}$ is Π^1_1, but, in fact, more is true: $\mathcal{O}$ is Π^1_1 complete, meaning that every Π^1_1 set reduces to $\mathcal{O}$ in a way analogous to every r.e. set reducing to $\emptyset'$.

Theorem 2.8 (Kleene; I.5.4 & I.5.5 in Sacks)**.** *$\mathcal{O}$ is uniformly many-one complete for Π^1_1 sets, and consequently, $\mathcal{O}$ is not Σ^1_1.*

The first part of this theorem says that if X is a Π^1_1 set, then there is a recursive function f such that $x \in X$ iff $f(x) \in \mathcal{O}$. The uniformity means that we can find an index for the function f computably from an index e such that $x \in X$ iff $(\forall f)(\exists y)T(f_y, e, x, y)$ where T is Kleene's predicate which is equivalent to $\Phi_e^{f_y}(x)$ halting in y many steps, and f_y is $f \upharpoonright \{n < y\}$. A key corollary of this fact is another result of Spector that is known as Σ^1_1 boundedness.

Corollary 2.9 (Spector [25]; I.5.6 in Sacks)**.** *Suppose $X \subseteq \mathcal{O}$ is Σ^1_1, then there is an $a \in \mathcal{O}$ such that $|x| \leq |a|$ for all $x \in X$.*

Furthermore, for future applications it is helpful to have a unique notation for each ordinal. In light of the Σ^1_1 boundedness theorem, the simplest we could hope for such a set to be is Π^1_1.

Theorem 2.10 (Feferman & Spector [7]; III.2.4 in Sacks)**.** *There is a Π^1_1 set $\mathcal{O}_1$ such that the restriction of $<_{\mathcal{O}}$ to $\mathcal{O}_1$ is linear, and every constructive ordinal has a notation in $\mathcal{O}_1$.*

The above definitions and results relativise to a set Z to give $\Pi^1_1(Z)$ sets $\mathcal{O}^Z$ and $<_{\mathcal{O}^Z}$ (you allow Z-recursive sequences in the limit case) which give rise to the Z-constructive ordinals.

2.2 Effective transfinite recursion

The definition of $|\cdot|$ was by induction on $\mathcal{O}$. We would like to define various things by induction on $\mathcal{O}$, but we would also like some control over the complexity of the object we build. It may seem far-fetched to hope that we could, say, define a function on $\mathcal{O}$ by induction and get something recursive (how would you store the data at a transfinite stage?), and indeed, if you insist that the domain is precisely $\mathcal{O}$ you cannot construct a recursive function. If, however, you allow the function to be defined on other members of ω, then we can get by.

Theorem 2.11 (I.3.2 in Sacks)**.** *Let $<_R$ be a wellfounded relation whose field is a subset of ω, and let $h : \omega \to \omega$ be a recursive function such that for all indices e and all x in the field of $<_R$ if $\Phi_e(y)$ is defined for all $y <_R x$, then $\Phi_{h(e)}(x)$ is defined. Then there is an e such that $\Phi_e(x)$ is defined for all x in the field of R and $\Phi_e = \Phi_{h(e)}$.*

The proof of this theorem uses the recursion theorem: If you pick a fixed point of h, then the hypothesis implies that $\Phi_{h(e)}$ is defined on the field of $<_R$. The method of applying this theorem is called **effective transfinite recursion** along $<_R$. Using effective transfinite recursion you can, for instance, define a map $+_{\mathcal{O}}$ such that $a, b \in \mathcal{O}$ iff $a +_{\mathcal{O}} b \in \mathcal{O}$ and if $a, b \in \mathcal{O}$, then $|a| + |b| = |a +_{\mathcal{O}} b|$.

The trick to using the theorem is to use the different cases that define $\mathcal{O}$ (0, successor, and limit) without worrying whether or not the number you are operating on is actually a notation. For instance we can define a recursive function

$h(e)$ such that

$$\Phi_{h(e)}(a,b) = \begin{cases} a & \text{if } b = 1; \\ 2^{\Phi_e(a,c)} & \text{if } b = 2^c; \\ ??? & \text{if } b = 3 \cdot 5^d; \\ 7 & \text{otherwise.} \end{cases}$$

Deliberately leaving the 3rd case mysterious for the moment, note that the first two cases mirror the definition of ordinal addition given that 1 represents 0 and 2^b represents the successor.

The third case is slightly trickier, we want

$$\begin{aligned} \left|a +_{\mathcal{O}} 3 \cdot 5^d\right| &= |a| + \left|3 \cdot 5^d\right| \\ &= |a| + \left|\lim_{n\to\infty} \Phi_d(n)\right| \\ &= \lim_{n\to\infty} \left(|a| + |\Phi_d(n)|\right) \\ &= \lim_{n\to\infty} \left|a +_{\mathcal{O}} \Phi_d(n)\right|. \end{aligned}$$

Uniformly in a, d, and an index e for $+_{\mathcal{O}}$ we can find an index $g(e,a,d)$ such that $\Phi_{g(e,a,d)}(n) = \Phi_e(a, \Phi_d(n))$, so in the third case we let $\Phi_{h(e)}(a,b) = 3 \cdot 5^{g(e,a,d)}$. An application of the recursion theorem to h and induction on $\mathcal{O}$ proves that a fixed point for h gives a computable function $+_{\mathcal{O}}$ with the desired properties.

We have defined an ordinal as constructive if it is represented by some notation. A different notion of effective ordinal is that it is isomorphic to some recursive relation.

Definition 2.12. An ordinal is **recursive** if it is either finite, or it is isomorphic to some recursive wellordering of ω.

Note that if a binary relation on ω is recursive (or even r.e.), then its field is

r.e. Furthermore, if the field of an r.e. relation is infinite, then there is a recursive bijection between the field and ω (you map 0 to the first thing enumerated, 1 to the second, and so on). So, if an r.e. relation is a wellordering of its field, then we can pull that relation through the recursive bijection to get an r.e. wellordering of ω. Because the original relation is a wellordering, eventually $x < y$ or $y < x$ gets enumerated for each x and y in the field. Hence, the isomorphic copy of the relation on ω is recursive, because we can wait to see which of $x < y$ or $y < x$ gets enumerated to determine which pairs are not in the relation. In summary, recursive wellorderings of ω give the same order-types as r.e. wellorderings of an r.e. set.

One can show, by effective transfinite recursion, that $\{b \in \mathcal{O} : b <_{\mathcal{O}} a\}$ and $\{(b, c) : b <_{\mathcal{O}} c <_{\mathcal{O}} a\}$ are (uniformly) r.e. in a. As the first of these sets is wellordered by the second, then $|a|$ is recursive as well as constructive. The converse is also true.

Theorem 2.13 (Kleene; I.4.4 in Sacks)**.** *An ordinal is recursive if and only if it is constructive.*

The recursive ordinals are relativised to a set Z by allowing Z-recursive wellorderings of ω. The Z-recursive ordinals are also equal to the Z-constructive ordinals.

2.3 The hyperarithmetic sets and hyperarithmetic reducibility

The hyperarithmetic sets are defined as the downward closure under $<_T$ of the iterates of the Turing jump through the recursive ordinals.

Definition 2.14 (H-sets)**.** We define a family of sets by recursion on $\mathcal{O}$:

$$H_1 = \emptyset, \quad H_{2^a} = (H_a)', \quad \text{and} \quad H_{3\cdot 5^e} = \bigoplus_{n\in\omega} H_{\Phi_e(n)}.$$

A set is **hyperarithmetic** if it is Turing reducible to some H-set. We denote the class of all hyperarithmetic sets by HYP.

It is clear by induction on $b \in \mathcal{O}$ that if $a <_{\mathcal{O}} b$, then $H_a \leq_T H_b$; indeed this is true uniformly (i.e., there is a recursive function $g(a,b)$ such that $\Phi_{g(a,b)}^{H_b} = H_a$ for each $a <_{\mathcal{O}} b \in \mathcal{O}$). This fact is a theorem of Spector's. However, even more is true: we don't need a, b to be $<_{\mathcal{O}}$ comparable:

Theorem 2.15 (Spector [25]; II.4.5 in Sacks)**.** *There is a recursive function $h(a,b)$ such that if $a, b \in \mathcal{O}$ and $|a| \leq |b|$, then $H_a = \Phi_{h(a,b)}^{H_b}$.*

An easy corollary is Spector's uniqueness theorem:

Corollary 2.16 (Spector [25]; II.4.6 in Sacks)**.** *If $a, b \in \mathcal{O}$ and $|a| = |b|$, then $H_a \equiv_T H_b$.*

The uniqueness theorem and the set of unique ordinal notations $\mathcal{O}_1$ allows us to define $\emptyset^{(\alpha)}$, the α-th jump of the empty set, as H_a where $a \in \mathcal{O}_1$ and $|a| = \alpha$. The natural hierarchy on the hyperarithmetic sets allows us to prove things about them by induction. This is particularly helpful because of the following equivalence:

Theorem 2.17 (Kleene; II.2.5 in Sacks)**.** *A set is hyperarithmetic if and only if it is* Δ^1_1.

So we can prove things about Δ^1_1 sets by induction along the hyperarithmetic hierarchy. These theorems relativise to Z by defining the H^Z-sets (the difference is that $H_1 = Z$ and you continue through $\mathcal{O}^Z$). Closing the H^Z-sets downward under Turing reducibility gives the Z-hyperarithmetic sets, denoted $\mathrm{HYP}(Z)$.

Proposition 2.18 (Shoenfield; II.5.2 in Sacks)**.** *For each* $n \geq 1$ *the relation "*X *is* $\Delta^1_n(Y)$*" is transitive.*

Clearly X is $\Delta^1_n(X)$ for each X, and so, as with Turing and arithmetic reducibility, there is a notion of Δ^1_n degree with an associated notion of reducibility.

Definition 2.19 (Hyperarithmetic reducibility)**.** We write $X \leq_h Y$ if X is hyperarithmetic in Y, i.e., if $X \in \mathrm{HYP}(Y)$. The **hyperarithmetic degrees** $\mathcal{D}_h$ are the quotient of 2^ω under the equivalence relation $X \equiv_h Y$ if and only if $X \leq_h Y$ and $Y \leq_h X$.

The degree of the hyperarithmetic sets is the least degree in $\mathcal{D}_h$. The operation $\oplus$ defined in Section 1.1 descends to $\mathcal{D}_h$ and endows it with the structure of a USL with least element.

2.4 The hyperjump

The **hyperjump** of X is $\mathcal{O}^X$. The hyperjump is somewhat like the Turing jump: They hyperjump of X is a $\Pi^1_1(X)$ complete set, and the Turing jump of X is a $\Sigma^0_1(X)$ complete set. The following proposition implies that the hyperjump is well defined on hyperdegrees:

Proposition 2.20 (II.7.1 in Sacks)**.** *$X \leq_h Y$ if and only if $\mathcal{O}^X \leq_m \mathcal{O}^Y$.*

As a point of contrast between the hyperarithmetic and the recursive, the Σ^0_1 Turing degrees form a complex substructure of $\mathcal{D}_T$, whereas there are only two Π^1_1 hyperdegrees.

Proposition 2.21 (Spector [25]; II.7.2 in Sacks)**.** *If X is Π^1_1, then X is either hyperarithmetic or $X \equiv_h \mathcal{O}$.*

Recall that ω_1^{CK} is the least nonconstructive ordinal and ω_1^X is the least X-nonconstructive ordinal. Of course, ω_1^X equals the height of $\mathcal{O}^X$ for each X. It is natural to ask how complicated a set must be so that $\omega_1^X > \omega_1^{\mathrm{CK}}$.

Proposition 2.22 (Spector [25]; II.7.3 & II.7.4 in Sacks)**.** *For all sets $X, Y \subseteq \omega$ we have*

(1) $X \leq_h Y$ implies $\omega_1^X \leq \omega_1^Y$;

(2) $\mathcal{O}^X \leq_h Y$ implies $\omega_1^X < \omega_1^Y$;

(3) $\omega_1^X < \omega_1^Y$ and $X \leq_h Y$ implies $\mathcal{O}^X \leq_h Y$.

We note two important application of this proposition.

Corollary 2.23. *For each X, $\omega_1^{CK} < \omega_1^X$ iff $\mathcal{O} \leq_h X$, and if $X <_h \mathcal{O}$, then $\mathcal{O}^X \equiv_h \mathcal{O}$.*

Proof. For the first claim observe that (2) of Proposition 2.22 with $X = \emptyset$ and $Y = X$ implies that

$$\mathcal{O} \equiv_h \mathcal{O}^\emptyset \leq_h X \Rightarrow \omega_1^{\mathrm{CK}} = \omega_1^\emptyset < \omega_1^X,$$

and (3) of the same Proposition implies that

$$\omega_1^{\mathrm{CK}} = \omega_1^{\emptyset} < \omega_1^X \text{ and } \emptyset \leq_h X \Rightarrow \mathcal{O} \equiv_h \mathcal{O}^{\emptyset} \leq_h X.$$

As $\emptyset$ is hyperarithmetic in every set X, we have our equivalence.

For the second claim, suppose that $X <_h \mathcal{O}$. By the first claim we know $\omega_1^X \equiv_h \omega_1^{\mathrm{CK}}$ and we are assuming $X <_h \mathcal{O}$, hence, an application of (3) yields that $\mathcal{O}^X \leq_h \mathcal{O}$. Monotonicity of the hyperjump tells is that $\mathcal{O} \equiv_h \mathcal{O}^{\emptyset} \leq_h \mathcal{O}_X$, and so $\mathcal{O}^X \equiv_h \mathcal{O}$. □

2.5 Cohen forcing

Paul Cohen invented the method of forcing to show that the negation of the continuum hypothesis is consistent with ZFC. His method has been expanded upon greatly in set theory and has also been adapted for use in recursion theory. Feferman [6] adapted Cohen forcing to the hyperarithmetic setting.

We start by constructing a language and a model to force over. More details can be found in Sacks III.4

Definition 2.24 ($\mathcal{L}(\omega_1^{\mathrm{CK}}, \mathtt{G})$)**.** We define a two-sorted language $\mathcal{L}(\omega_1^{\mathrm{CK}}, \mathtt{G})$. The first-order primitive terms are numerals $\underline{0}, \underline{1}, \underline{2}, \ldots$ and variables. We have symbols $\underline{+}$ and $\underline{\times}$ to represent $+$ and $\times$, respectively. The first-order terms are build in the standard way.

The second-order primitive terms include a symbol $\mathtt{G}$, unranked set variables, and ranked set variables $X^\beta, Y^\beta, \ldots$ for each $\beta < \omega_1^{\mathrm{CK}}$. We also have a binary relation $\underline{\in}$ between first-order and second-order objects to represent $\in$.

The full language $\mathcal{L}(\omega_1^{\mathrm{CK}}, \mathtt{G})$ is built in the standard way from atomic formulas $t = s$ and $t \in U$ for t, s first-order terms and U a second-order term via the propositional collectives and quantifiers over all three kinds of variables.

Definition 2.25 (Rank of a formula)**.** A formula φ of $\mathcal{L}(\omega_1^{\mathrm{CK}}, \mathtt{G})$ is **ranked** if all its set variables are ranked. The **(ordinal) rank** of a ranked formula is the least α such that $\beta \leq \alpha$ for each X^β occurring freely in φ, and $\beta < \alpha$ for all X^β occurring bound in φ. A formula is Σ^1_1 if it is ranked or of the form $(\exists X_1) \cdots (\exists X_n)\psi$ for some ranked formula ψ.

The **full ordinal rank** of a ranked sentence φ is a function

$$r : [-1, \omega_1^{\mathrm{CK}}] \to \omega$$

such that for each $\beta < \omega_1^{\mathrm{CK}}$, $r(\beta)$ is the number of occurrences of $(\exists X^\beta)$ or $(\forall X^\beta)$ (over all variables of rank β). $r(-1)$ is the number of first-order quantifiers in ψ. Clearly r is finitely supported. If r_1, r_2 are full ordinal ranks, then we say $r_1 < r_2$ if at the last place β where r_1, r_2 disagree $r_1(\beta) < r_2(\beta)$.

It can be shown that the $<$ relation between full ordinal ranks is a wellordering of height ω_1^{CK}.

We would like to assign Gödel numbers to the formulas of our language in a manner that is as effective as possible. Given that we quantify over ω_1^{CK} to define the ranked variables, the best we could hope for is Π^1_1. Recall the set $\mathcal{O}_1$ of unique ordinal notations for members of ω_1^{CK}; Numbering X^β by $\langle 2, b\rangle$ for the unique notation $b \in \mathcal{O}_1$ for β, and numbering everything else in any standard way renders a Gödel numbering such that the set of Gödel numbers is Π^1_1 and for each recursive α, the set of Gödel numbers of ranked formulas of rank less than α is r.e. uniformly in the notation for α in $\mathcal{O}_1$. It is convenient to assume that formulas are in prenex

normal form. The standard manipulations to change a formula into prenex normal form does not increase full ordinal rank, and is recursive on Gödel numbers.

Definition 2.26 (The ramified analytic hierarchy)**.** We define a model to go along with the language $\mathcal{L}(\omega_1^{\mathrm{CK}}, \mathtt{G})$. The first-order part will always be ω and we will interpret the arithmetic symbols by their true interpretation. Consequently, our model is specified in terms of what subset of 2^ω the second-order quantifiers range over. By a simultaneous induction on $\beta < \omega_1^{\mathrm{CK}}$ we define structures $\mathcal{M}(\beta, G)$ and truth of sentences of rank at most β in $\bigcup_{\alpha<\beta} \mathcal{M}(\beta, G)$.

If φ is a sentence of rank at most β, then any bound variable X^α appearing in φ satisfies $\alpha < \beta$. φ is true in $\bigcup_{\alpha<\beta} \mathcal{M}(\alpha, G)$ if φ is true with $\mathtt{G}$ is interpreted as G, X^α is restricted to $\mathcal{M}(\alpha, G)$, the first-order variables range over ω, and everything else has its usual meaning.

$\mathcal{M}(\beta, G)$ is the collection of sets defined as follows: Let $\psi(x)$ be a ranked formula of rank at most β with only x free. Define $\hat{x}\psi(x)$ to be the set of all n such that $\psi(n)$ is true in $\bigcup_{\alpha<\beta} \mathcal{M}(\alpha, G)$. Then $\hat{x}\psi(x)$ is a member of $\mathcal{M}(\beta, T)$.

$\mathcal{M}(\omega_1^{\mathrm{CK}}, G) = \bigcup_{\alpha<\omega_1^{\mathrm{CK}}} \mathcal{M}(\alpha, G)$, and we define truth of a sentence φ in $\mathcal{M}(\omega_1^{\mathrm{CK}}, G)$ by allowing unranked set quantifiers to range over $\mathcal{M}(\omega_1^{\mathrm{CK}}, G)$ and the rest of the symbols are interpreted as above.

The following key lemma justifies our interest in the ramified analytic hierarchy.

Lemma 2.27 (III.4.16 in Sacks)**.** *The following are equivalent:*

(1) $\omega_1^{CK} = \omega_1^G$*;*

(2) $\mathcal{M}(\omega_1^{CK}, G) = HYP(G)$*;*

(3) $\mathcal{M}(\omega_1^{CK}, G)$ *satisfies* Δ_1^1 *comprehension.*

Where Δ^1_1 comprehension is the axiom scheme

$$(\forall x)[(\exists Y)A(x,Y) \leftrightarrow (\forall Z)B(x,Z)] \rightarrow (\exists X)(\forall x)[x \in X \leftrightarrow (\exists Y)A(x,Y)]$$

for arithmetic predicates A and B. The Δ^1_1 comprehension scheme says that if a set is Δ^1_1 definable in a model, then that set exists in the model.

This equivalence will be very helpful when we construct sets by forcing. Say we want to force a set G to satisfy some fact related to $\leq_h$, for instance, that it is not hyperarithmetically above X. A natural way to try to do this would be to try to diagonalize against every set below $G^{(\alpha)}$ for each G-recursive α. However, to do this we would need to know ahead of time what ordinals will be recursive in G. If we can guarantee that $\mathcal{M}(\omega_1^{\mathrm{CK}}, G)$ satisfies Δ^1_1 comprehension, then we will know what ordinals are recursive in G, and because $\mathrm{HYP}(G) = \mathcal{M}(\omega_1^{\mathrm{CK}}, G)$ in this case, and every member of $\mathcal{M}(\omega_1^{\mathrm{CK}}, G)$ is named by a term in the language $\mathcal{L}(\omega_1^{\mathrm{CK}}, \mathtt{G})$, then we can diagonalize by using our forcing language.

Definition 2.28 (Cohen forcing)**. Cohen forcing** is the set of all finite binary strings $2^{<\omega}$ ordered by extension, i.e., $\sigma \leq \tau$ if $|\sigma| \geq |\tau|$ and $\sigma \restriction_{|\tau|} = \tau$, i.e., if $\sigma \supseteq \tau$. We think of a string as an approximating a set by specifying an initial segment of its characteristic function. The **forcing relation** $\Vdash$ is between binary strings and sentences of $\mathcal{L}(\omega_1^{\mathrm{CK}}, \mathtt{G})$ and is defined by induction on the full ordinal rank and logical complexity of sentences as follows (note that t, t_1 and t_2 are variable free first-order terms, we assume that $\forall$ and $\vee$ are abbreviations for the dual of $\exists$ and $\wedge$, and for a formula $\varphi(\mathbf{X}^\alpha)$ and a formula $\psi(x)$ of rank at most α with only x free we define $\varphi(\hat{x}\psi(x))$ to be the formula obtained from φ by replacing every instance of $t \in \mathbf{X}^\alpha$ with $\psi(t)$. This replacement lowers full ordinal rank.)

- $\sigma \Vdash \underline{n} \in \mathtt{G}$ iff $\sigma(n) \downarrow = 1$;

- $\sigma \Vdash t \in \mathtt{G}$ iff $\sigma \Vdash \underline{n} \in \mathtt{G}$ for the n that t represents under the standard interpretation of the symbols of arithmetic;
- $\sigma \Vdash t_1 = t_2$ iff t_1 and t_2 represent the same integer under the standard interpretation of the symbols of arithmetic;
- $\sigma \Vdash \varphi \wedge \psi$ iff $\sigma \Vdash \varphi$ and $\sigma \Vdash \psi$;
- $\sigma \Vdash (\exists x)\varphi(x)$ iff there exists n such that $\sigma \Vdash \varphi(\underline{n})$;
- $\sigma \Vdash (\exists X^\alpha)\varphi(X^\alpha)$ iff $\sigma \Vdash \varphi(\hat{x}\psi(x))$ for some $\psi(x)$ of rank at most α;
- $\sigma \Vdash (\exists X)\varphi(X)$ iff there exists $\alpha < \omega_1^{\mathrm{CK}}$ such that $\sigma \Vdash (\exists X^\alpha)\varphi(X^\alpha)$;
- $\sigma \Vdash \neg\varphi$ iff for all $\tau \supseteq \sigma$, $\tau \nVdash \varphi$.

A set G is **(Cohen) generic** with respect to a sentence φ if there exists some string $\sigma \subset G$ such that either $\sigma \Vdash \varphi$ or $\sigma \Vdash \neg\varphi$ (we say σ **decides** φ). X is generic if it is generic with respect to every sentence. A sequence $\{\sigma_n\}_{n\in\omega}$ of strings is generic if $\sigma_n \subseteq \sigma_{n+1}$ for all n, and for each sentence φ there is an n such that σ_n decides φ.

If a set G is generic, then the sequence of initial segments of G is a generic sequence. Similarly, if a sequence $\{\sigma_n\}$ is generic, then $G = \{x : (\exists n)\sigma_n(x) \downarrow = 1\}$ is a generic set.

Next is a sequence of basic facts about forcing. Each time we introduce a new notion of forcing we will have to prove similar facts.

Proposition 2.29 (IV.3.4 & IV.3.5 in Sacks)**.** *The Cohen forcing relation satisfies the following facts for all strings σ and sentences φ:*

- *(consistency) σ does not force both φ and $\neg\varphi$;*

- *(density) there exists $\tau \supseteq \sigma$ deciding φ;*
- *(extension preserves forcing) if $\sigma \Vdash \varphi$ and $\tau \supseteq \sigma$, then $\tau \Vdash \varphi$.*

Furthermore, if G is a generic, then truth equal forcing, i.e.,

$$\mathcal{M}(\omega_1^{CK}, G) \Vdash \varphi \quad \textit{iff} \quad (\exists \sigma \subset G)[\sigma \Vdash \varphi].$$

The proof of the first two claims is, essentially, the definition of forcing for negation. The final two are proved by induction on the full ordinal rank and logical complexity of φ. As $\mathcal{L}(\omega_1^{\mathrm{CK}}, \mathtt{G})$ is countable, it follows that generic sets exist, indeed continuum many generics exist. Another key fact is that there is a generic (strictly) hyperarithmetic in $\mathcal{O}$.

Theorem 2.30 (Feferman [6]; IV.3.6 in Sacks)**.** *If G is generic, then Δ^1_1 comprehension holds in $\mathcal{M}(\omega_1^{CK}, G)$.*

It can be shown that if G is generic, then so too are the columns $G^{[n]}$ of G. Furthermore, the columns are **very independent**, i.e.,

$$G^{[n]} \not\leq_h \bigoplus_{i \neq n} G^{[i]}, \quad \text{and} \quad \bigoplus_{i \neq n} G^{[i]} \not\leq_h G^{[n]}.$$

To relativise Cohen forcing to a set Z you add a symbol $\mathtt{Z}$ to the language, and continue it through the Z-recursive ordinals for $\mathcal{L}(\omega_1^Z, \mathtt{G}, \mathtt{Z})$. The model $\mathcal{M}(\omega_1^Z, G, Z)$ is defined similarly to $\mathcal{M}$ with $\mathtt{Z}$ interpreted as Z. The forcing relation is modified to add a clause for sentences of the form $t \in \mathtt{Z}$, which are forced iff they are true. A generic for this forcing and language will preserve ω_1^Z, i.e. $\omega_1^{G \oplus Z} = \omega_1^Z$ and the model will be all sets hyperarithmetic in $G \oplus Z$. The columns of a generic will be very independent over Z, i.e., they are very independent even when you join both sides with Z.

2.6 The theory of the hyperdegrees

Recall that $\mathcal{D}_h$ is a USL with least element and a notion of jump. One of the ways we look to understand this structure is through what kind of substructures can be embedded within $\mathcal{D}_h$ and, stepping up in complexity, when can embeddings be extended to larger substructures.

Cohen forcing suffices to show that every countable partial order embeds in $\mathcal{D}_h$, even below $\mathcal{O}$. This shows that a Σ_1 sentence is true in $(\mathcal{D}_h, \leq_h)$ iff it is consistent with the theory of partial orders; whether a sentence is consistent with the theory of partial orders is a computable question therefore the Σ_1 theory of $(\mathcal{D}_h, \leq_h)$ is decidable.

A Π_2 sentence $(\forall \overline{x})(\exists \overline{y})\varphi$ can be interpreted as asking about when embeddings extend; no matter how you choose the $\overline{x}$ to satisfy φ can you choose corresponding $\overline{y}$. This intuition is complicated by the fact that φ might not fully determine all the relationships between the $\overline{x}$ and the $\overline{y}$. However, over the theory of USLs you can transform a sentence in the language $\{\sqsubseteq, \sqcup, \bot\}$ into questions of the form "For fixed finite USLs $\mathscr{U}$ and $\mathscr{V}_1, \ldots, \mathscr{V}_n$ where each $\mathscr{V}_i$ is an extension of $\mathscr{U}$, does every embedding of $\mathscr{U}$ into $\mathcal{D}_h$ extend to one of the $\mathscr{V}_i$?" This equivalence is Theorem B.1 in Appendix B.

We call this question the (finite) extension of embeddings problem for $\mathcal{D}_h$ as a USL. If we can answer these questions, then we can decide the Π_2 theory of $\mathcal{D}_h$ as an USL with least element. We are also concerned with the corresponding extension of embeddings problem for $\mathcal{D}_h(\leq_h \mathcal{O})$ as a $\mathrm{USL}^\top$. It is shown in Theorem B.2 that answering this question for a $\mathrm{USL}^\top$ suffices to decide its Σ_2 theory.

Our answer to the extension of embeddings problem for $\mathcal{D}_h$ is that we can

always extend an embedding of a finite USL $\mathscr{U}$ to a finite extension $\mathscr{V}$ iff $\mathscr{U}$ is an initial segment of $\mathscr{V}$, and in $\mathcal{D}_h(\leq_h \mathcal{O})$ we can always extend an embedding of a finite USL$^\top$ $\mathscr{U}$ to a finite extension $\mathscr{V}$ iff $\mathscr{U}$ is an almost initial segment of $\mathscr{V}$, i.e., the only element of $\mathscr{U}$ above any $v \in \mathscr{V} \setminus \mathscr{U}$ is $\top$. Consequently, the Π_2 theories of both of these structures in the corresponding languages is decidable.

To prove these results we need to provide a method to extend embeddings when $\mathscr{U}$ is an (almost) initial segment, and to show that there is an embedding of $\mathscr{U}$ that does not extend when $\mathscr{U}$ is not an (almost) initial segment. We call the first half of this the positive half, and it is the subject of Chapter 3. The second half is called the negative half and is covered in Chapter 4. Both of these are forcing constructions. Along the way we also consider the extension of embeddings problem for the arithmetic degrees and the countable extension of embeddings question in $\mathcal{D}_h$.

CHAPTER 3

EXTENDING USL EMBEDDINGS IN THE HYPERDEGREES

In this chapter we prove the following theorem, which can be thought of as the "positive half" of the solution to the extension of embeddings problem for $\mathcal{D}_h$.[1]

Theorem 3.1. *If $\mathscr{U}$ is a finite USL and $\mathscr{V}$ is a finite end extension of $\mathscr{U}$, then every embedding of $\mathscr{U}$ into $\mathcal{D}_h$ extends to an embedding of $\mathscr{V}$.*

We prove this theorem in two special cases: where $\mathscr{V}$ is a free extension of $\mathscr{U}$ and $\mathscr{V}$ is a simple end extension of $\mathscr{U}$ (definitions can be found in Appendix A). These results suffice to prove the more general case because every finite end extension of $\mathscr{U}$ is a subUSL of a simple end extension of a free extension of $\mathscr{U}$. This algebraic result is due to Jockusch and Slaman [10]. For completeness, it is given as Theorem A.1 which appears in Appendix A.

3.1 Free extensions

Firstly, free extensions:

Theorem 3.2. *Suppose $\mathscr{U}$ is a finite USL and $\mathscr{V}$ is the free extension of $\mathscr{U}$ by a finite set X. Then every embedding of $\mathscr{U}$ into $\mathcal{D}_h$ extends to an embedding of $\mathscr{V}$.*

Proof. Suppose $\mathscr{V}$ is a free extension of $\mathscr{U}$ by n many new generators $\{g_1, \ldots, g_n\}$ (i.e., $X = \{g_1, \ldots, g_n\}$). Further, suppose that $f : \mathscr{U} \to \mathcal{D}_h$ is a USL embedding of $\mathscr{U}$ into $\mathcal{D}_h$. Let $\mathbf{a} \in \mathcal{D}_h$ be a degree hyperarithmetically above every degree in the image of f, and let G be a Cohen generic relative to $\mathbf{a}$.

[1] Most of this chapter's work appears in Barnes [2]

Recall from the discussion proceeding Theorem 2.30 that the columns of G are all strongly independent over $\mathbf{a}$, i.e., if $i \in \omega$, then

$$G^{[i]} \not\leq_h A \oplus \bigoplus_{j \neq i} G^{[j]}$$

where A is a set of degree $\mathbf{a}$. Furthermore, G is hyperarithmetically incomparable with $\mathbf{a}$. Consequently, we extend f from $\mathscr{U}$ to $\mathscr{V}$ by defining

$$\tilde{f}(v) = f(u) \oplus \bigoplus_{i \in I} G^{[i]}, \text{ where } v = u \sqcup \bigsqcup_{i \in I} g_i \text{ for } I \subseteq \{1, \ldots, n\}.$$

This map is well-defined as every element of v has a unique expression as the join of some $u \in \mathscr{U}$ and some subset of $\{g_1, \ldots, g_n\}$. It extends f as the subset of $\{g_1, \ldots, g_n\}$ corresponding to a $u \in \mathscr{U}$ is the emptyset.

To show that $\tilde{f}$ is an embedding it suffices to show it is injective, preserves $\perp_{\mathscr{V}}$, and preserves $\sqcup_{\mathscr{V}}$. Clearly, as $\tilde{f}$ extends f which is an USL embedding of $\mathscr{U}$, $\tilde{f}$ maps $\perp_{\mathscr{U}}$ to the degree of the hyperarithmetic sets. As $\perp_{\mathscr{V}} = \perp_{\mathscr{U}}$, the map preserves least element.

If $\tilde{f}(v_1) = \tilde{f}(v_2)$, then writing $v_1 = u_1 \sqcup \bigsqcup_{i \in I_1} g_i$ and $v_2 = u_2 \sqcup \bigsqcup_{i \in I_2} g_i$ it follows that

$$f(u_1) \oplus \bigoplus_{i \in I_1} G^{[i]} \equiv_h f(u_2) \oplus \bigoplus_{i \in I_2} G^{[i]}.$$

As $\mathbf{a}$ was chosen to be above the image of f, we know that $f(u_1), f(u_2) \leq_h A$, and consequently

$$f(u_1) \oplus \bigoplus_{i \in I_1} G^{[i]} \leq_h f(u_2) \oplus \bigoplus_{i \in I_2} G^{[i]} \leq_h A \oplus \bigoplus_{i \in I_2} G^{[i]}.$$

As the columns of G are strongly independent over $\mathbf{a}$, it follows that $I_1 \subseteq I_2$. Interchanging 1 and 2 gives the reverse inclusion, and so $I_1 = I_2$. Furthermore, as $A \geq_h f(u_1), f(u_2)$, yet $A \not\leq_h G$, it also follows that $f(u_1) \equiv_h f(u_2)$, and so, as f is an embedding, $u_1 = u_2$ which shows that $\tilde{f}$ is injective.

Finally we want to show that $\tilde{f}$ preserves $\sqcup$. By definition of $\tilde{f}$ we have that

$$\begin{aligned}
\tilde{f}(v_1) \oplus \tilde{f}(v_2) &\equiv_h f(u_1) \oplus \bigoplus_{i \in I_1} G^{[i]} \oplus f(u_2) \oplus \bigoplus_{i \in I_2} G^{[i]} \\
&\equiv_h f(u_1) \oplus f(u_2) \oplus \bigoplus_{i \in I_1 \cup I_2} G^{[i]} \\
&\equiv_h f(u_1 \sqcup u_2) \oplus \bigoplus_{i \in I_1 \cup I_2} G^{[i]} \\
&\equiv_h \tilde{f}\left((u_1 \sqcup u_2) \sqcup \bigsqcup_{i \in I_1 \cup I_2} g_i \right) \\
&\equiv_h \tilde{f}\left((u_1 \sqcup u_2) \sqcup \bigsqcup_{i \in I_1} g_i \sqcup \bigsqcup_{i \in I_2} g_i \right) \\
&\equiv_h \tilde{f}(v_1 \sqcup v_2).
\end{aligned}$$

Hence, $\tilde{f}$ is a USL embedding of $\mathscr{V}$ extending f as required. □

The reader may observe that the strong independence of a Cohen real actually allows you to extend an embedding from a countable USL $\mathscr{U}$ by countably many free generators.

3.2 Simple end extensions

Now we turn to simple end extensions. A free extension by a single generator g is a kind of simple end extension; It is the free extension with the fewest positive join facts in the sense that if $a \sqsubseteq b \sqcup g$ in the extension, then $a \sqsubseteq b$ in the original USL. Our plan for the full class of simple end extensions is to reduce even further to the case where you allow one new join fact to hold. The proof of the sufficiency of this reduction is given in Appendix A Theorem A.3.

Theorem 3.3. *Let* $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}_i$, *and* $\boldsymbol{d}_i$ *be hyperarithmetic degrees satisfying*

$$\bigwedge_{i\in\omega} \boldsymbol{d}_i \not\leq_h \boldsymbol{c}_i \,\&\, (\boldsymbol{a} \not\leq_h \boldsymbol{c}_i \text{ or } \boldsymbol{d}_i \not\leq_h \boldsymbol{c}_i \oplus \boldsymbol{b}).$$

Then for every collection of hyperarithmetic degrees $\boldsymbol{e}_j$ *there is a degree* g *such that*

$$\boldsymbol{b} \leq_h \boldsymbol{a} \oplus g \,\&\, \bigwedge_{i\in\omega} \boldsymbol{d}_i \not\leq_h \boldsymbol{c}_i \oplus g \,\&\, \bigwedge_{i\in\omega} g \not\leq_h \boldsymbol{e}_i.$$

The proof of this theorem is a relatively involved forcing construction.

3.2.1 The notion and language of forcing

Definition 3.4 (Turing functionals)**.** For $x \in \omega, y \in \{0,1\}$, and $\sigma \in 2^{<\omega}$, we call the tuple $\langle x, y, \sigma\rangle$ an **axiom** or a **computation**. A **Turing functional** is a set Φ of computations such that if σ_1 and σ_2 are compatible and $\langle x, y_1, \sigma_1\rangle, \langle x, y_2, \sigma_2\rangle \in \Phi$, then $y_1 = y_2$ and $\sigma_1 = \sigma_2$.

Furthermore, Φ is **use-monotone** if

- whenever $\sigma_1 \subsetneq \sigma_2$ and $\langle x_1, y_1, \sigma_1\rangle, \langle x_2, y_2, \sigma_2\rangle \in \Phi$, then $x_1 < x_2$, and
- whenever $x_1 < x_2$ and $\langle x_2, y_2, \sigma_2\rangle \in \Phi$, there is a $y_1 \in \{0,1\}$ and a $\sigma_1 \subsetneq \sigma_2$ such that $\langle x_1, y_1, \sigma_1\rangle \in \Phi$.

Notation. We write $\Phi(x, \sigma) = y$ to mean there is a $\tau \subseteq \sigma$ such that $\langle x, y, \tau\rangle \in \Phi$. If $X \subseteq \omega$, we write $\Phi(x, X) = y$ to mean there is an n such that $\Phi(x, X \upharpoonright n) = y$, and we write $\Phi(X)$ for the (partial) function evaluated in this way. Note that $\Phi(X)$ is (uniformly) recursive in the join of X and Φ.

Definition 3.5 (Kumabe-Slaman forcing)**. Kumabe-Slaman forcing** is the partial order $\mathcal{P}$ with conditions $p = (\Phi_p, \mathbf{X}_p)$, where Φ_p is a finite use-monotone Turing

functional, and $\mathbf{X}_p$ is a finite set of subsets of ω (hereafter we refer to such $X \subseteq \omega$ as **reals**). For elements p, q of $\mathcal{P}$, we say $q \leq_{\mathcal{P}} p$ (q **extends** p) if

- $\Phi_p \subseteq \Phi_q$, and for all $(x_q, y_q, \sigma_q) \in \Phi_q \setminus \Phi_p$ and all $\langle x_p, y_p, \sigma_p \rangle \in \Phi_p$, the length of σ_q is greater than the length of σ_p;
- $\mathbf{X}_p \subseteq \mathbf{X}_q$; and
- for every x, y and every $X \in \mathbf{X}_p$, if $\Phi_q(X)(x) = y$, then $\Phi_p(X)(x) = y$, i.e., q can't add new axioms $\langle x, y, \sigma \rangle$ where $\sigma \subset X$ for any $X \in \mathbf{X}_p$.

We think of a forcing condition $p = (\Phi_p, \mathbf{X}_p)$ as approximating a (possibly non-recursive) Turing functional Φ_G. To a descending sequence of conditions $\{p_n\}_{n=1}^{\infty}$ we associate the object $\Phi_G = \bigcup_n \Phi_{p_n}$, a use-monotone Turing functional.

On a formal level, we code each finite use-monotone Turing functional Φ as a natural number, which encodes each element of Φ and also the size of Φ. For instance, given some recursive encoding of triples $\langle x, y, \sigma \rangle$ as positive integers, we could code $\Phi = \{\langle x_1, y_1, \sigma_1 \rangle, \ldots, \langle x_n, y_n, \sigma_n \rangle\}$ as the product $\prod_{i=1}^{n} P(i)^{\langle x_i, y_i, \sigma_i \rangle}$ where $P(i)$ is the ith prime. Consequently, a forcing condition is coded as a pair: the first coordinate is a natural number and the second a finite set of reals. We dub the first coordinate the **finite part** and the second the **infinite part**. Given finite use-monotone Turing functionals Φ_p and Φ_q, we will occasionally abuse notation and write $\Phi_p \leq_{\mathcal{P}} \Phi_q$, by which we mean $(\Phi_p, \emptyset) \leq_{\mathcal{P}} (\Phi_q, \emptyset)$.

We will often have occasion to modify our notion of forcing somewhat, either by insisting that certain axioms don't belong to the finite part, or that the reals can only come from a particular subclass of 2^{ω}. For the first modification we introduce some notation:

Notation. Given reals A and B (think of representatives from $\mathbf{a}$ and $\mathbf{b}$, respectively), we say that a Turing functional Φ **partially computes** B **on input** A if whenever $\Phi(A)(n)$ is defined, it equals $B(n)$. More explicitly, if $\langle x, y, \sigma \rangle \in \Phi$ is an axiom such that $\sigma \subseteq A$ (we say this axiom **applies to** A), then $y = B(x)$.

Definition 3.6 (Restricted Kumabe-Slaman forcing)**.** $\mathcal{Q}_{A,B}$ is the subset of $\mathcal{P}$ containing all those conditions $q = (\Phi_q, \mathbf{X}_q)$ such that Φ_q partially computes B on input A and $\mathbf{X}_q$ does not contain A. Extension in $\mathcal{Q}_{A,B}$ (denoted $\leq_{\mathcal{Q}_{A,B}}$) is defined as for $\mathcal{P}$. The partial order $\mathcal{Q}_{A,B}$ is a suborder of $\mathcal{P}$ and will be abbreviated as $\mathcal{Q}$ if it is understood what A and B are. We employ a similar convention as for $\mathcal{P}$ and write $\Phi_q \leq_{\mathcal{Q}} \Phi_r$ to mean $(\Phi_q, \emptyset) \leq_{\mathcal{Q}} (\Phi_r, \emptyset)$.

Our goal is to construct a sufficiently generic sequence of conditions into which we code B via A. More precisely, for each $n \in \omega$, our generic will have an axiom $(n, B(n), \alpha)$ for some α that is an initial segment of A. This will ensure that $\Phi_G(A)$ is the characteristic function of B, and so $B \leq_T \Phi_G \oplus A$. We will need to show that we can construct a generic sequence without adding "incorrect" axioms applying to A, and without adding A into the infinite part of any condition (which would prevent us from adding any axioms about A later).

The forcing $\mathcal{Q}_{A,B}$ has all this hard-wired in; however, it is at the expense of checking whether Φ can serve as the finite part of the condition being recursive only in $A \oplus B$, instead of plain old recursive. For $\mathcal{P}$, we need to show that we can manually avoid adding in bad axioms to the finite part or adding A into the infinite part of a condition, while still constructing a generic sequence.

We ensure that $\mathbf{d}_i \not\leq \mathbf{c}_i \sqcup g$ via genericity. At a high level, our method is quite standard: we will define a forcing language that will be powerful enough to express all the possible pertinent reductions from $C_i \oplus \Phi_G$ to D_i (for chosen representatives

$C_i \in \mathbf{c}_i$ and $D_i \in \mathbf{d}_i$). We will try to diagonalize against each of these reductions in turn.

For forcing in the arithmetic or Turing degrees, a forcing language based upon first-order arithmetic often suffices. However, for the hyperarithmetic setting something more expressive is required. We use recursive infinitary formulas as presented in Ash and Knight [1] Chapter 9 (over the language of arithmetic, see Shore [20] for an exposition). Roughly speaking, at the ground level we have the quantifier-free finitary sentences in our language, and then, for each recursive ordinal α, a Σ^r_α formula is an r.e. disjunction of the form

$$\bigvee_i (\exists \vec{u}_i)\psi_i(\vec{u}_i, \vec{x})$$

where each ψ_i is a $\Pi^r_{\beta_i}$ formula for some $\beta_i < \alpha$. Similarly, a Π^r_α formula is an r.e. conjunction of universally quantified formulas all of which are $\Sigma^r_{\beta_i}$ for some $\beta_i < \alpha$.

Definition 3.7 (The language of forcing)**.** We start with $\mathcal{L}$, the set containing the following nonlogical symbols:

- Numerals: $\underline{0}, \underline{1}, \underline{2}, \ldots$,
- a 5-ary relation for the universal recursive predicate: $\varphi(e, x, s, y, \sigma)$ (to be interpreted as follows: the eth Turing machine on input x running for s steps with oracle σ halts and outputs y; we will often write this as $\{e\}^\sigma_s(x) \downarrow = y$, and write $\{e\}^X(x)$ for the partial function evaluated this way),
- a predicate for a string being an initial segment of our generic: $\sigma \subset \Phi_{\mathtt{G}}$.

From $\mathcal{L}$ we build up the recursive infinitary formulas as indicated in Ash and Knight, with the convention that the ground level formulas are in CNF, and denote this collection $\mathcal{L}^r_{\omega_1,\omega}$.

To relativise to some $S \subseteq \omega$, we add a predicate (written $\sigma \subset \mathtt{S}$) for a string being an initial segment of S, and build up the S-recursive infinitary formulas, i.e., we allow S-r.e. infinitary conjunctions and disjunctions. We denote the collection of such formulas $\mathcal{L}^r_{\omega_1,\omega}(S)$ and rank the formulas as $\Sigma^r_\alpha(S)$ and $\Pi^r_\alpha(S)$ for $\alpha < \omega_1^S$.

We think of numerals as having a dual role in our language: they represent both numbers (in the obvious way) and also strings via some fixed recursive bijection between $2^{<\omega}$ and ω. We will suppress the technical apparatus needed, and will use lowercase Roman letters when we are thinking in terms of numbers, and lowercase Greek letters when we are thinking in terms of strings.

We can now define the forcing relation $\Vdash_{\mathcal{P}}$ for $\mathcal{P}$, with some $S \subseteq \omega$ as our set parameter.

Definition 3.8 (The forcing relation)**.** The relation $\Vdash_{\mathcal{P}}$ between elements of $\mathcal{P}$ and sentences of $\mathcal{L}^r_{\omega_1,\omega}(S)$ in is defined by induction on the ordinal rank and complexity of the formula.

(1) $p \Vdash_{\mathcal{P}} \underline{n} = \underline{m}$ iff $n = m$,

(2) $p \Vdash_{\mathcal{P}} \varphi(\underline{e}, \underline{x}, \underline{s}, \underline{y}, \underline{\sigma})$ iff $\{e\}^\sigma_s(x) \downarrow = y$,

(3) $p \Vdash_{\mathcal{P}} \underline{\sigma} \subset \mathtt{S}$ iff $\sigma \subset S$,

(4) $p \Vdash_{\mathcal{P}} \underline{\sigma} \subset \Phi_{\mathtt{G}}$ iff for all n less than $|\sigma|$:

  (a) if $\sigma(n) = 1$, then $n \in \Phi_p$,

  (b) if $\sigma(n) = 0$, then either n is not of the form $\langle x, y, \tau\rangle$, or $n = \langle x, y, \tau\rangle$, $n \notin \Phi_p$ and

    (i) there is an $\langle x_0, y_0, \tau_0\rangle \in \Phi_p$ such that $|\sigma_0|$ is greater than $|\sigma|$, or, x_0 is greater than x and σ_0 is compatible with σ, or

(ii) σ is an initial segment of one of the elements of $\mathbf{X}_p$,

(5) For an atomic sentence ψ, $p \Vdash_{\mathcal{P}} \neg\psi$ iff there is no $q \leq_{\mathcal{P}} p$ with $q \Vdash \psi$,

(6) For sentences $\psi_1, \ldots, \psi_n$ that are literals, $p \Vdash_{\mathcal{P}} \psi_1 \vee \cdots \vee \psi_n$ iff for all $q \leq_{\mathcal{P}} p$ there exists an $r \leq_{\mathcal{P}} q$ and an $i \in \{1, \ldots, n\}$ such that $r \Vdash \psi_i$,

(7) For sentences $\psi_1, \ldots, \psi_n$ that are finite disjunctions of literals, $p \Vdash_{\mathcal{P}} \psi_1 \wedge \cdots \wedge \psi_n$ iff $p \Vdash \psi_i$ for each $i \in \{1, \ldots, n\}$,

(8) $p \Vdash_{\mathcal{P}} \bigvee_i (\exists \vec{u}_i)\psi_i(\vec{u_i})$ iff there is some i and some $\vec{n}$ such that $p \Vdash_{\mathcal{P}} \psi_i(\underline{\vec{n}})$,

(9) $p \Vdash_{\mathcal{P}} \bigwedge_i (\forall \vec{u}_i)\psi_i(\vec{u_i})$ iff for each $i, \vec{n}$ and $q \leq_{\mathcal{P}} p$ there is some $r \leq_{\mathcal{P}} q$ such that $r \Vdash \psi_i(\underline{\vec{n}})$.

Note. We note some abuses of the language of forcing which we will employ hereafter. Firstly, we will stop underlining numerals. Secondly, although our language only has negations of atomic formulas, there is an S-recursive function neg that takes (an index for) a formula ψ and returns (an index for) a formula that is logically equivalent to the negation of ψ. Also, neg preserves the ordinal rank of the formula, i.e., if ψ is $\Pi^r_\alpha(S)$, then $\text{neg}(\psi)$ is $\Sigma^r_\alpha(S)$, and vice versa. Consequently, we write $\neg\psi$ freely as if negation were a part of the language.

As expected, it can be shown that for each S-recursive ordinal α, the claim that the partial function $\{e\}^{(S \oplus \Phi_G)^{(\alpha)}}$ is defined on an input x (and, optionally, has value y) can be expressed as a $\Sigma^r_{\alpha+1}$ sentence in our language, uniformly in e, x, and y.

The definition of $\Vdash_{\mathcal{Q}}$ is analogous except in (4). In particular, we must add a third possibility to (b), namely, that $\tau \subset A$ but $B(x) \neq y$.

Standard lemmas include that extension preserves forcing, no condition forces

a sentence and its (formal) negation, and given a sentence and a condition, we can find an extension of that condition deciding the sentence.

The countability of $\mathcal{L}^r_{\omega_1,\omega}(S)$ implies that generic sequences exist. To a generic sequence $\{p_n\}$ we associate the generic object $\Phi_G = \bigcup \Phi_{p_n}$, which is the intended interpretation of $\Phi_{\mathtt{G}}$. With this, one proves the final standard lemma: truth equals forcing. Proofs of these standard lemmas can be found in Chapter 10 of Ash and Knight [1].

Now we prove a technical lemma about the complexity of the forcing relation.

Lemma 3.9. *Given a finite use-monotone Turing functional Φ_0 and a quantifier-free finitary sentence ψ in $\mathcal{L}^r_{\omega_1,\omega}(S)$ (i.e., a Π^r_0 sentence), we can S-recursively decide whether there is an $\boldsymbol{X}$ such that $(\Phi_0, \boldsymbol{X}) \in \mathcal{P}$ forces ψ, uniformly in both the sentence and the functional.*

Furthermore, we can also make this decision for $\mathcal{Q}_{A,B}$ (and $\Vdash_{\mathcal{Q}}$); however, the procedure is uniformly recursive in $A \oplus B \oplus S$.

Proof. We begin with the claim for $\mathcal{P}$. By convention, finitary sentences are in conjunctive normal form. There are four types of atomic sentences

(1) $n = m$,

(2) $\varphi(e, x, s, y, \sigma)$,

(3) $\sigma \subset \mathtt{S}$,

(4) $\sigma \subset \Phi_{\mathtt{G}}$,

and there are the negations of these atoms, finite disjunctions of such literals, and finite conjunctions of such disjunctions.

For the first three types of atoms, a condition forces it iff it is true, and we can recursively in S check each of these facts. Similarly, if one of these atoms is false, then no condition forces it to be true, and so every condition forces the negation.

The fourth type of atom is $\sigma \subset \Phi_{\mathtt{G}}$. We claim that there is an $\mathbf{X}$ so that $(\Phi_0, \mathbf{X}) \Vdash \sigma \subset \Phi_{\mathtt{G}}$ iff for each $n < |\sigma|$, we have $\Phi_0(n) = \sigma(n)$ (which is uniformly recursive to check). To see this, note that a condition p forces $\sigma \subset \Phi_{\mathtt{G}}$ iff $\sigma(n) = \Phi_p(n)$ and, for each n such that $\sigma(n) = 0$, if $n = \langle x, y, \tau \rangle$, then $\langle x, y, \tau \rangle$ is prevented from entering any extension of p (i.e. there is either an axiom $(x_0, y_0, \tau_0) \in \Phi_p$ with $|\tau_0| > |\tau|$, or $x_0 \geq x$ and τ_0 and τ are compatible, or τ is an initial segment of some $X \in \mathbf{X}_p$).

Therefore, for p to force $\sigma \subset \Phi_{\mathtt{G}}$, it is necessary that Φ_p agrees with σ everywhere that σ is defined. If Φ_0 and σ do agree, then for each $n < |\sigma|$ with $\sigma(n) = 0$ and n of the form $\langle x, y, \tau \rangle$, we can pick a real X extending τ and let $\mathbf{X}$ be the collection of such reals. Then $(\Phi_0, \mathbf{X}) \Vdash \sigma \subset \Phi_{\mathtt{G}}$, which completes the claim.

Now we turn to literals of the form $\neg(\sigma \subset \Phi_{\mathtt{G}})$. We claim that we can pick an $\mathbf{X}$ to guarantee that $(\Phi_0, \mathbf{X}) \Vdash_{\mathcal{P}} \neg(\sigma \subset \Phi_{\mathtt{G}})$ iff there is an $n < |\sigma|$ such that $\sigma(n) \neq \Phi_0(n)$. From the definition of forcing, a condition p forces $\neg(\sigma \subset \Phi_{\mathtt{G}})$ iff every extension of p fails to force $\sigma \subset \Phi_{\mathtt{G}}$. Suppose Φ_0 and σ agree wherever σ is defined. Then, by the above, we can pick $\mathbf{X}_0$ so that $(\Phi_0, \mathbf{X}_0) \Vdash \sigma \subset \Phi_{\mathtt{G}}$. Therefore, for each $\mathbf{X}$ we have that $(\Phi_0, \mathbf{X} \cup \mathbf{X}_0) \leq_{\mathcal{P}} (\Phi_0, \mathbf{X})$ and $(\Phi_0, \mathbf{X} \cup \mathbf{X}_0) \Vdash \sigma \subset \Phi_{\mathtt{G}}$. Consequently, if Φ_0 and σ agree wherever σ is defined, then for each choice of $\mathbf{X}$ there is an extension of $(\Phi_0, \mathbf{X})$ forcing $\sigma \subset \Phi_{\mathtt{G}}$. By taking the contrapositive we have established one direction of the claim.

For the other direction, suppose there is an n so that $\sigma(n) \neq \Phi_0(n)$. For such

an n, either $\sigma(n) = 0$ and $\Phi_0(n) = 1$, or $\sigma(n) = 1$ and $\Phi_0(n) = 0$. In the first case, no extension of $(\Phi_0, \emptyset)$ can force $\sigma \subset \Phi_{\mathtt{G}}$ as n is in every stronger condition. Hence, $\mathbf{X} = \emptyset$ suffices as the set of reals. In the second case, if $n \neq \langle x, y, \tau \rangle$, then n is not an element of any Turing functional and so again $\mathbf{X} = \emptyset$ suffices. So suppose $n = \langle x, y, \tau \rangle$, and let X be any real extending τ. As $\langle x, y, \tau \rangle \notin \Phi_0$, it is also not in any extension of $(\Phi_0, \{X\})$, so no extension of this condition can force $\sigma \subset \Phi_{\mathtt{G}}$, and thus $\{X\}$ will work as our set of reals, which completes the proof of the claim.

Now suppose we have a sentence $\psi = l_1 \vee \cdots \vee l_n$, which is a finite disjunction of literals. From the definition of forcing, it is clear that if a condition forces one of the disjuncts, it forces the whole disjunction. So, using induction, ask whether there is a literal l_i and a set of reals $\mathbf{X}$ so that $(\Phi_0, \mathbf{X}) \Vdash l_i$. If there is, we are done, as such an $\mathbf{X}$ suffices. If not, we claim no $\mathbf{X}$ works. To see this, firstly note that we get to ignore any of the literals with atomic parts of type (1), (2) or (3), as $(\emptyset, \emptyset)$ decides these sentences as false. So now assume ψ is a disjunction of literals built from atoms of type (4). For each positive literal $\sigma_i \subset \Phi_{\mathtt{G}}$, because we cannot pick an $\mathbf{X}$ to force it, Φ_0 and σ_i disagree somewhere on the domain of σ_i. Hence, for each i, there is a finite collection $\mathbf{X}_i$ so that $(\Phi_0, \mathbf{X}_i) \Vdash \neg(\sigma_i \subset \Phi_{\mathtt{G}})$. Similarly, for each negative literal, as we can't pick an $\mathbf{X}$ to force it, Φ_0 and σ_i agree on the domain of σ_i, and so, we can pick $\mathbf{X}_i$ so that $(\Phi_0, \mathbf{X}_i) \Vdash \sigma_i \subset \Phi_{\mathtt{G}}$.

So fix a set of reals $\mathbf{X}$, and define $\mathbf{X}_0 = \bigcup_{i=1}^{n} \mathbf{X}_i$. Then $(\Phi_0, \mathbf{X}_0 \cup \mathbf{X}) \leq_{\mathcal{P}} (\Phi_0, \mathbf{X})$ yet $(\Phi_0, \mathbf{X}_0 \cup \mathbf{X})$ forces the opposite of each literal in ψ, i.e., forces $\neg\psi$. Consequently, $(\Phi_0, \mathbf{X})$ does not force ψ for any $\mathbf{X}$.

For conjunctions $\psi_1 \wedge \cdots \wedge \psi_m$, the definition of forcing is you force the conjunction iff you force each of the conjuncts. Inductively, ask if there are reals $\mathbf{X}_i$ such that $(\Phi_0, \mathbf{X}_i) \Vdash \psi_i$; if "yes" in each case, then $(\Phi_0, \bigcup_{i=1}^{m} \mathbf{X}_i)$ forces each conjunct.

If any of them return a "no', then no choice of reals can suffice for all conjuncts. This completes the decision procedure; uniformity is clear.

For $\mathcal{Q}$, only minor changes are needed. Firstly, one must check whether Φ_0 correctly computes B on input A, which is recursive in $A \oplus B$. Secondly, although the definition of $\Vdash_{\mathcal{Q}}$ for formulas of the form $\sigma \subset \Phi_{\mathtt{G}}$ is slightly different (i.e., an axiom $\langle x, y, \sigma \rangle$ where $\sigma \subset A$, yet $y \neq B(x)$ is not a member of any $\mathcal{Q}$ condition), this is again $A \oplus B$-recursive to check. Finally, whenever we picked a real X to prevent an axiom from entering any stronger condition, X just had to extend some finite string, and so need not be A. These are the only observations needed. $\square$

Now we break the proof of the extended Posner-Robinson theorem into cases. For the moment, imagine we are trying to do a single instance of our goal:

$$[\mathbf{d} \not\leq_h \mathbf{c} \,\&\, (\mathbf{a} \not\leq_h \mathbf{c} \text{ or } \mathbf{d} \not\leq_h \mathbf{b} \sqcup \mathbf{c})] \to (\exists g)[\mathbf{b} \leq_h \mathbf{a} \sqcup g \,\&\, \mathbf{d} \not\leq_h \mathbf{c} \sqcup g]$$

for some hyperarithmetic degrees $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{d}$. Suppose they satisfy the antecedent. Then either $\mathbf{a} \not\leq_h \mathbf{c}$, or $\mathbf{a} \leq_h \mathbf{c}$ and $\mathbf{d} \not\leq_h \mathbf{b} \sqcup \mathbf{c}$. We call the former the easy case, and the latter the hard case. Pick representatives A, B, C, and D from $\mathbf{a}, \mathbf{b}, \mathbf{c}$, and $\mathbf{d}$, respectively; furthermore, if we are in the hard case, ensure that $A \leq_T C$.

3.2.2 The easy case

We now proceed with the easy case, for which we use the forcing $\mathcal{P}$. While constructing our generic, we will need to code B into the join of A and Φ_G. Our coding strategy is to add axioms $\langle n, B(n), \alpha \rangle$ to our generic, with $\alpha \subset A$. To ensure that this is possible, we need to show that we can construct a generic sequence without

adding A to the infinite part of any condition and such that any axiom $\langle x, y, \sigma\rangle$ applying to A that we add to the finite part satisfies $B(x) = y$.

Additionally, we need to show that a sufficiently generic sequence ensures that $D \not\leq_h C \oplus \Phi_G$. Towards this goal we use the forcing language $\mathcal{L}^r_{\omega_1,\omega}(C)$. As our language can only express reductions from some C-recursive jump of $C \oplus \Phi_G$, we must also show that Φ_G preserves ω_1^C. Thus, we have three goals: coding B, diagonalizing against computing D, and preserving ω_1^C. In this section we work towards the first two goals, leaving the last for Section 5.

Coding B into the join

Definition 3.10 ($(\mathcal{P}, C)$-essential to $\neg\psi$ over Φ_0)**.** Let Φ_0 be a finite use-monotone Turing functional and

$$\psi = \bigwedge_i (\forall \vec{u}_i)\theta_i(\vec{u}_i)$$

a $\Pi^r_\alpha(C)$ sentence in $\mathcal{L}^r_{\omega_1,\omega}(C)$ for some $\alpha > 0$.

For a sequence $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_n)$ of elements of $2^{<\omega}$ all of the same length, we say that $\boldsymbol{\tau}$ is $(\mathcal{P}, C)$**-essential to** $\neg\psi$ **over** Φ_0 if for all $p \in \mathcal{P}$, all i, and all $\vec{m}$ of the correct length, if p is a condition such that $p <_{\mathcal{P}} (\Phi_0, \emptyset)$ and $p \Vdash_{\mathcal{P}} \neg\theta_i(\vec{m})$, then $\Phi_p \setminus \Phi_0$ includes a triple $\langle x, y, \sigma\rangle$ such that σ is compatible with at least one component of $\boldsymbol{\tau}$.

Definition 3.11 ($T_{\mathcal{P},C}(\Phi_0, \psi, k)$)**.** For each finite use-monotone Turing functional Φ_0, each $\Pi^r_\alpha(C)$ sentence ψ of our forcing language with $\alpha > 0$, and each natural number k, let $T_{\mathcal{P},C}(\Phi_0, \psi, k)$ be the set of length k vectors of binary strings all of the same length that are $(\mathcal{P}, C)$-essential to $\neg\psi$ over Φ_0.

We drop the $(\mathcal{P}, C)$ prefix or subscript where confusion will not arise (as in the rest of this section, where we have fixed C and are only dealing with $\mathcal{P}$).

We order $T(\Phi_0, \psi, k)$ by extension on all coordinates. Being essential is closed downward in this order. This endows $T(\Phi_0, \psi, k)$ with the structure of a subtree of the length k vectors of binary sequences of equal length, so $T(\Phi_0, \psi, k)$ is a subtree of a recursively bounded recursive tree.

Lemma 3.12. *Suppose that Φ_0 is a finite use-monotone Turing functional, ψ is a $\Pi^r_\alpha(C)$ sentence in our forcing language with $\alpha > 0$, and k is a natural number.*

(1) If there is a size k set $\boldsymbol{X}$ of reals such that $(\Phi_0, \boldsymbol{X}) \Vdash \psi(\Phi_G)$, then $T(\Phi_0, \psi, k)$ is infinite.

(2) If $T(\Phi_0, \psi, k)$ is infinite, then it has an infinite path Y. Further, each such Y is naturally identified with a size k set $\boldsymbol{X}(Y)$ of reals such that $(\Phi_0, \boldsymbol{X}(Y)) \Vdash \psi(\Phi_G)$.

Proof. For the first claim, let $\mathbf{X}$ be such a set, and let $(X_1, \ldots, X_k)$ be an enumeration of it. Further, let $\boldsymbol{\tau}_l$ denote $(X_1 \upharpoonright l, \ldots, X_k \upharpoonright l)$. We show that each $\boldsymbol{\tau}_l \in T(\Phi_0, \psi, k)$, proving the tree is infinite.

Suppose $(\forall \vec{u}_i)\theta_i(\vec{u}_i)$ is one of the conjuncts that makes up ψ, $\vec{m}$ is the same length as $\vec{u}_i$, and $p \leq_{\mathcal{P}} (\Phi_0, \emptyset)$ forces $\neg\theta_i(\vec{m})$, so consequently, $p \Vdash \neg\psi$. As p forces $\neg\psi$ and $(\Phi_0, \mathbf{X})$ forces ψ, these two conditions are incompatible in $\mathcal{P}$, and as $p \leq_{\mathcal{P}} (\Phi_0, \emptyset)$ there must be an axiom $\langle x, y, \sigma\rangle \in \Phi_p$ which is forbidden from entering any extension of $(\Phi_0, \mathbf{X})$. But then σ is an initial segment of some $X_i \in \mathbf{X}$, and so σ is compatible with the corresponding component of $\boldsymbol{\tau}_l$, showing $\boldsymbol{\tau}_l$ is essential.

For the second claim, suppose $T(\Phi_0, \psi, k)$ is infinite. As it is a subtree of a finitely branching tree, König's lemma provides an infinite path Y through it. To each such Y, associate $\mathbf{X}(Y)$, the componentwise union of the coordinates of the members of Y. This will be a size k set of reals. We claim $(\Phi_0, \mathbf{X}(Y))$ forces ψ for each such Y.

Suppose $p < (\Phi_0, \emptyset)$, and there is an i and an $\vec{m}$ such that $p \Vdash \neg\theta_i(\vec{m})$. Then there is an axiom $\langle x, y, \sigma\rangle \in \Phi_p \setminus \Phi_0$ where σ is compatible with some component of each element of Y. In particular, for sufficiently large elements of Y, σ is an initial segment of such a component and so is an initial segment of some $X \in \mathbf{X}(Y)$. Then, p does not extend $(\Phi_0, \mathbf{X}(Y))$. Therefore, no extension of $(\Phi_0, \mathbf{X}(Y))$ forces $\neg\theta_i(\vec{m})$ for any i and $\vec{m}$, and as each extension of $(\Phi_0, \mathbf{X}(Y))$ has a further extension deciding $\theta_i(\vec{m})$, by the definition of forcing, $(\Phi_0, \mathbf{X}(Y)) \Vdash \psi$. □

Notation. For a set $S \subset \omega$, we say a decision procedure or property of natural numbers is $\Pi^0_\alpha(S)$ for an S-recursive ordinal $\alpha > 0$ if the set of solutions to the procedure or the property, respectively, is co-r.e. in $S^{(\alpha)}$ for $\alpha \geq \omega$ and co-r.e. in $S^{(\alpha-1)}$ for finite α. We say it is $\Delta^0_\alpha(S)$ if it and its complement are both $\Pi^0_\alpha(S)$. For example, a $\Pi^0_1(S)$ set is co-r.e. in S, but a $\Pi^0_{\omega+1}(S)$ set is co-r.e. in $S^{(\omega+1)}$.

Lemma 3.13. *For each finite use-monotone Turing functional Φ_0, each $\Pi^r_\alpha(C)$ sentence ψ with $\alpha > 0$, and each number k, $T(\Phi_0, \psi, k)$ is $\Pi^0_\alpha(C)$ uniformly in Φ_0, ψ and k.*

Proof. Fix a length k vector $\boldsymbol{\tau}$ of binary strings of the same length. Recall that $\boldsymbol{\tau}$ is essential to $\neg\psi$ over Φ_0 if for all $p = (\Phi_p, \mathbf{X}_p) <_{\mathcal{P}} (\Phi_0, \emptyset)$, all conjuncts $(\forall\vec{u})\theta_i(\vec{u})$ which make up ψ, and all $\vec{m}$ of the correct length, if $p \Vdash \neg\theta_i(\vec{m})$, then $\Phi_p \setminus \Phi_0$ contains an axiom $\langle x, y, \sigma\rangle$ where σ is compatible with some component of $\boldsymbol{\tau}$.

We can express this, equivalently, as

> For all $\Phi_p \leq_{\mathcal{P}} \Phi_0$, all $\vec{m}$ of the correct length, and all conjuncts $(\forall \vec{u})\theta_i(\vec{u})$ which make up ψ, if there exists a set $\mathbf{X}$ such that $(\Phi_p, \mathbf{X}) \Vdash \neg\theta_i(\vec{m})$, then $\Phi_p \setminus \Phi_0$ contains an axiom $\langle x, y, \sigma\rangle$ where σ is compatible with some component of $\boldsymbol{\tau}$.

Or more compactly as

$$(\forall \Phi_p)(\forall \vec{m})(\forall i)[((\Phi_p, \emptyset) \leq_{\mathcal{P}} (\Phi_0, \emptyset) \;\&\; (\exists \mathbf{X})(\Phi_p, \mathbf{X}) \Vdash \neg\theta_i(\vec{m})) \\ \rightarrow ((\exists \langle x, y, \sigma\rangle \in \Phi_p \setminus \Phi_0) \text{ with } \sigma \text{ compatable with some } \tau \in \boldsymbol{\tau})].$$

We can also "quantify out" the size of the set $\mathbf{X}$, as

$$(\forall \Phi_p)(\forall \vec{m})(\forall j, i)[((\Phi_p, \emptyset) \leq_{\mathcal{P}} (\Phi_0, \emptyset) \;\&\; (\exists \mathbf{X}, |\mathbf{X}| = j)(\Phi_p, \mathbf{X}) \Vdash \neg\theta_i(\vec{m})) \\ \rightarrow ((\exists \langle x, y, \sigma\rangle \in \Phi_p \setminus \Phi_0) \text{ with } \sigma \text{ compatable with some } \tau \in \boldsymbol{\tau})].$$

We use both of these formulations in an inductive proof of this lemma, the first in the base case, and the second in the inductive steps. Note that in both cases, the $(\forall i)$ is a quantifier over a C-r.e. set: the conjuncts making up ψ. This is still equivalent to a single universal quantifier.

Suppose $\alpha = 1$. In this case, for each i and $\vec{m}$ of the correct length, $\neg\theta_i(\vec{m})$ is a quantifier-free sentence. So, by Lemma 3.9, deciding whether there is an $\mathbf{X}$ such that $(\Phi_p, \mathbf{X}) \Vdash \neg\theta_i(\vec{m})$ is uniformly C-recursive in Φ_p, θ_i, and $\vec{m}$. Also, checking whether $\Phi_p \leq_{\mathcal{P}} \Phi_0$ and whether there is an axiom $\langle x, y, \sigma\rangle \in \Phi_p \setminus \Phi_0$ compatible with some $\tau \in \boldsymbol{\tau}$ is uniformly recursive in Φ_0, Φ_p and $\boldsymbol{\tau}$.

Consequently, in the case $\alpha = 1$, $\boldsymbol{\tau}$ being essential to $\neg\theta$ over Φ_0 can be expressed as a property with two universal natural number quantifiers, then a

universal quantifier over a C-r.e. set, then a matrix which is C-recursive uniformly in all parameters. Such an expression is co-r.e. in C, or $\Pi^0_1(C)$ as required.

If $\alpha > 1$ is a successor, we can assume all of its conjuncts are $\Sigma^r_{\alpha-1}(C)$ by Ash and Knight. By Lemma 3.12, there being an $\mathbf{X}$ of size j such that $(\Phi_p, \mathbf{X}) \Vdash \neg\theta_i(\vec{m})$ is equivalent to $T(\Phi_p, \neg\theta_i(\vec{m}), j)$ being infinite. But $T(\Phi_p, \neg\theta_i(\vec{m}), j)$ is, by induction, a $\Pi^0_\alpha(C)$ subtree of a recursively bounded recursive tree. Hence, it being infinite is also a $\Pi^0_\alpha(C)$ fact, given uniformly in Φ_p, j, θ_i, and $\vec{m}$. This bound follows because the tree being finite is equivalent to it being of bounded height, and our trees are subtrees of recursive trees. Hence, saying our tree is finite is equivalent to saying there is a level of a recursive tree, from which our tree is disjoint, i.e., $(\exists n)$[for all nodes x in the recursive tree T at level n, x is not in our tree]. If our tree is $\Pi^0_\alpha(C)$ for some C, this formula is equivalent to a $\Sigma^0_\alpha(C)$ formula, and so its negation is $\Pi^0_\alpha(C)$ as required.

Thus for successor α, $\boldsymbol{\tau}$ being essential to $\neg\psi$ over Φ_0 is equivalent to a property which we can express with a block of universal natural number quantifiers, then a universal quantifier over a C-r.e. set, then a $\Sigma^0_{\alpha-1}(C)$ matrix uniformly in the parameters. This shows the property is $\Pi^0_\alpha(C)$ as required.

For the limit case the argument is similar. The same formulation of the definition of $\boldsymbol{\tau}$ being essential works as for the successor case. However, here we have the full power of $C^{(\alpha)}$ which for $\beta < \alpha$, can decide whether recursively branching $C^{(\beta)}$-recursive trees are infinite, uniformly in an index for the tree, so $C^{(\alpha)}$ can uniformly decide the matrix. The prefix is a block of universal quantifiers, then a universal quantifier over a C-r.e. set, with uniformly $C^{(\alpha)}$-recursive matrix, so $\boldsymbol{\tau}$ being essential is uniformly co-r.e. in $C^{(\alpha)}$, or $\Pi^0_\alpha(C)$ as required. □

Corollary 3.14. *Suppose that $S \subseteq \omega$ is not $\Delta^0_\alpha(C)$. Let Φ_0 be a finite use-*

monotone Turing functional, ψ a $\Pi^r_\alpha(C)$ sentence with $\alpha > 0$, and $k \geq 1$. If there is a size k set $\boldsymbol{X}$ of reals such that $(\Phi_0, \boldsymbol{X}) \Vdash \psi$, then there is such a set not containing S. Moreover, we can find such an $\boldsymbol{X}$ all of whose members are recursive in $C^{(\alpha+1)}$ uniformly in ψ, k, and $S \oplus C^{(\alpha+1)}$.

Proof. Suppose there is a size k set $\mathbf{X}$ such that $(\Phi_0, \mathbf{X}) \Vdash \psi$. By Lemmas 3.12 and 3.13 the tree $T(\Phi_0, \psi, k)$ is a $\Pi^0_\alpha(C)$ subtree of a recursively bounded recursive tree which has an infinite path. The paths form a nonempty $\Pi^0_\alpha(C)$ class, and so there is a path Y in which S is not recursive. Then, certainly, S is not a member of $\mathbf{X}(Y)$. Further, Lemma 3.12 implies $(\Phi_0, \mathbf{X}(Y)) \Vdash \psi$. This completes the proof of the first claim.

For the second claim, we need to find an infinite path in $T(\Phi_0, \psi, k)$ on which S does not appear, recursively in $C^{(\alpha+1)} \oplus S$. We need $C^{(\alpha+1)}$ to decide which members of $T(\Phi_0, \psi, k)$ have an infinite part of $T(\Phi_0, \psi, k)$ above it, and S to decide whether a given node could possibly have an extension on which S appears. Consequently, there is an algorithm, recursive in $S \oplus C^{(\alpha+1)}$, solving this problem which searches through the tree of length k sequences of binary strings of the same length, looking for one which disagrees with S and has an infinite part of $T(\Phi_0, \psi, k)$ above it. Once one is found, the algorithm switches to searching for any path in $T(\Phi_0, \psi, k)$ through this node. Note the uniformity in ψ and k follows from the uniformity in the construction of $T(\Phi_0, \psi, k)$.

Of course, once we have found a node each of the components of which are not compatible with S, yet has a path through it, we never use S again in the construction. Hence, we could externally hard code to only search for paths in $T(\Phi_0, \psi, k)$ through some fixed node, and any such path Y corresponds to reals $\mathbf{X}(Y)$ which are recursive in $C^{(\alpha+1)}$ (as $C^{(\alpha+1)}$ is constructing such a Y). Hence, the

members of the $\mathbf{X}$ we build are $C^{(\alpha+1)}$-recursive although we lose the uniformity by externally hard-coding in some information. $\square$

Corollary 3.15. *Suppose that $\alpha > 0$, S is not $\Delta^0_\alpha(C)$, and ψ is a $\Pi^r_\alpha(C)$ sentence. For any condition $p = (\Phi_p, \boldsymbol{X}_p)$ with $S \notin \boldsymbol{X}_p$, there is a stronger q which we can find uniformly in Φ_p and $S \oplus C^{(\alpha+1)} \oplus \boldsymbol{X}_p$ such that*

(1) $S \notin \boldsymbol{X}_q$ and each $X \in \boldsymbol{X}_q \setminus \boldsymbol{X}_p$ is recursive in $C^{(\alpha+1)}$,

(2) for all x, if $\Phi_q(S)(x)$ is defined, then $\Phi_p(S)(x)$ is defined. That is, q does not add any new computations which apply to S,

(3) either $q \Vdash \psi$ or there is a conjunct θ_i and an $\vec{m}$ such that $q \Vdash \neg\theta_i(\vec{m})$ (and we can tell which formula we have forced).

Proof. Fix a condition $p = (\Phi_p, \mathbf{X}_p)$, and let $k = |\mathbf{X}_p|$. We would like to know whether $T(\Phi_0, \psi, k+1)$ is infinite. It is a $\Pi^0_\alpha(C)$ subtree of a recursively bounded recursive tree, and hence this is a $\Pi^0_\alpha(C)$ fact. If $T(\Phi_0, \psi, k+1)$ is infinite, then Corollary 3.14 provides a path Y through it, not containing S, each member of which is recursive in $S \oplus C^{(\alpha+1)}$. Then $(\Phi_p, \mathbf{X}_p \cup \mathbf{X}(Y)) \leq_{\mathcal{P}} (\Phi_p, \mathbf{X})$, $(\Phi_p, \mathbf{X}_p \cup \mathbf{X}(Y)) \Vdash \psi$, and, clearly, (2) is satisfied, as we haven't changed the finite part.

If $T(\Phi_0, \psi, k+1)$ is not infinite, then (enumerating $\mathbf{X} = \{X_1, \ldots, X_k\}$) $(X_1, \ldots, X_k, S)$ does not provide a path through it. Consequently, for some l, $(X_1 \upharpoonright l, \ldots, X_n \upharpoonright l, S \upharpoonright l)$ is not essential to $\neg\psi$ over Φ_p, meaning there is a conjunct θ_i, numerals $\vec{m}$, and a condition $q = (\Phi_q, \mathbf{X}_q)$ extending $(\Phi_p, \emptyset)$ such that $q \Vdash_{\mathcal{P}} \neg\theta_i(\vec{m})$, yet every new axiom $\langle x, y, \sigma\rangle \in \Phi_q \setminus \Phi_p$ is incompatible with each coordinate of $(X_1 \upharpoonright l, \ldots, X_n \upharpoonright l, S \upharpoonright l)$. In particular, there are no new axioms applying to S, nor to any $X_i \in \mathbf{X}$, and by Lemma 3.12 there is a j such that $T(\Phi_q, \neg\theta_i(\vec{m}), j)$ is infinite.

We can find which $\Phi_q < \Phi_p$ adds neither new computations applying to S, nore new computations to any member of $\mathbf{X}_p$, recursively in $S \oplus \mathbf{X}_p$, and then whether $T(\Phi_q, \neg\theta_i(\vec{m}), j)$ is infinite recursively in $C^{(\alpha)}$. Consequently, we can find such a $\Phi_q < \Phi_p, \theta_i, \vec{m}$, and j recursively in $S \oplus C^{(\alpha+1)} \oplus \mathbf{X}_p$ (we add the extra jump because for each $\Phi_q < \Phi_p$ we ask whether there exists a $\theta_i, \vec{m}$, and j such that $T(\Phi_p, \neg\theta_i(\vec{m}), j)$ is infinite). Once we have found such a $\Phi_q, \theta_i, \vec{m}$, and j, applying Corollary 3.15 again, there is a size j set $\mathbf{X}$ with $S \notin \mathbf{X}$ such that every $X \in \mathbf{X}$ is uniformly recursive in $S \oplus C^{(\alpha+1)}$ and $(\Phi_q, \mathbf{X}) \Vdash \neg\theta_i(\vec{m})$, hence $(\Phi_q, \mathbf{X} \cup \mathbf{X}_p)$ is the condition we want. □

As A is not $\Delta^0_\alpha(C)$ for any C-recursive α, we can repeatedly apply Corollary 3.15 to construct an $\mathcal{L}^r_{\omega_1,\omega}(C)$-generic sequence in $\mathcal{P}$, without adding A to the infinite part of any condition, and without adding any axioms applying to A. Hence, we can dovetail this construction with one which adds axioms $\langle n, B(n), \alpha\rangle$ for some sufficiently long $\alpha \subset A$ which effectively codes B into the join of A and Φ_G. Now we proceed with showing that we can diagonalize against $C \oplus \Phi_G$ computing D.

Diagonalizing against D

Notation. For a real S, we say $\langle e, \alpha\rangle$ is an S-hyperarithmetic reduction if e is an index and α is an S-recursive ordinal. We write $\langle e, \alpha\rangle^S$ for the partial function $\{e\}^{S^{(\alpha)}}$.

Definition 3.16 ($(\mathcal{P}, C)$-essential to splits)**.** Let Φ_0 be a finite use-monotone Turing functional and $\langle e, \alpha\rangle$ a C-hyperarithmetic reduction. For each finite sequence $\boldsymbol{\tau}$ of binary strings of the same length, we say $\boldsymbol{\tau}$ is **$(\mathcal{P}, C)$-essential to $\langle e, \alpha\rangle$-splits below** Φ_0 if whenever $p, q <_{\mathcal{P}} (\Phi_0, \emptyset)$ form an $\langle e, \alpha\rangle$-split (i.e., there is an x such

that $p \Vdash \langle e, \alpha \rangle^{C \oplus \Phi_G}(x) \downarrow = k_1, q \Vdash \langle e, \alpha \rangle^{C \oplus \Phi_G}(x) \downarrow = k_2$ and $k_1 \neq k_2$), there is some new axiom $\langle x, y, \sigma \rangle \in \Phi_p \cup \Phi_q$ but not in Φ_0 such that σ is compatible with some component of $\boldsymbol{\tau}$.

We let $U_{\mathcal{P},C}(\Phi_0, \langle e, \alpha \rangle, k)$ be the set of $\boldsymbol{\tau}$ of length k which are $(\mathcal{P}, C)$-essential to $\langle e, \alpha \rangle$-splits below Φ_0, and consider this as a tree as before. Also, as before, we drop the $\mathcal{P}$ and C when confusion will not arise.

Lemma 3.17. *Let Φ_0 be a Turing functional, $\langle e, \alpha \rangle$ a C-hyperarithmetic reduction, and k a natural number.*

- *If $\boldsymbol{X}$ is a size k set of reals such that no pair $p, q <_{\mathcal{P}} (\Phi_0, \boldsymbol{X})$ form an $\langle e, \alpha \rangle$-split, then $U(\Phi_0, \langle e, \alpha \rangle, k)$ is infinite.*
- *If $U(\Phi_0, \langle e, \alpha \rangle, k)$ is infinite, then it has an infinite path, and each such path Y is identified with a size k set $\boldsymbol{X}(Y)$ such that $(\Phi_0, \boldsymbol{X}(Y))$ has no $\langle e, \alpha \rangle$-splits below it.*

Proof. Let $\boldsymbol{\tau}_l = (X_1 \upharpoonright l, \ldots, X_k \upharpoonright l)$, where $X_1, \ldots, X_k$ enumerates $\mathbf{X}$. Suppose $p, q <_{\mathcal{P}} (\Phi_0, \emptyset)$ form an $\langle e, \alpha \rangle$-split. As $(\Phi_0, \mathbf{X})$ has no splits below it, at least one of p and q are not compatible with $(\Phi_0, \mathbf{X})$. Suppose it is p which is incompatible. As $p <_{\mathcal{P}} (\Phi_0, \emptyset)$, p must have a new axiom $\langle x, y, \sigma \rangle$ which some X_i forbids. Hence $\sigma \subset X_i$ and, therefore, σ is compatible with the ith component of $\boldsymbol{\tau}_l$. Hence, every $\boldsymbol{\tau}_l$ is essential to $\langle e, \alpha \rangle$-splits, and so $U(\Phi_0, \langle e, \alpha \rangle, k)$ is infinite.

For the second claim, suppose $U(\Phi_0, \langle e, \alpha \rangle, k)$ is infinite. Then Konig's lemma provides us a path Y, and $\mathbf{X}(Y)$ is the componentwise union of Y. Suppose $p, q \leq_{\mathcal{P}} (\Phi_0, \mathbf{X}(Y))$ form an $\langle e, \alpha \rangle$-split below $(\Phi_0, \mathbf{X}(Y))$; then they also form a split below $(\Phi_0, \emptyset)$. As the elements of Y are essential, there is some new axiom

$\langle x, y, \sigma \rangle$ in (WLOG) Φ_p, but not in Φ_0, compatible with some component of each element of Y. Hence σ is an initial segment of some component of $\mathbf{X}(Y)$, and so $p \not\leq_{\mathcal{P}} (\Phi_0, \mathbf{X}(Y))$, a contradiction. $\square$

Lemma 3.18. *If Φ_0 is a Turing functional, $\langle e, \alpha \rangle$ a C-hyperarithmetic reduction, and k a natural number, then $U(\Phi_0, \langle e, \alpha \rangle, k)$ is $\Pi^0_{\alpha+3}(C)$ uniformly in Φ_0, e, and k.*

Proof. Fix $\Phi_0, \langle e, \alpha \rangle, k$, and $\boldsymbol{\tau}$ of length k. Then $\boldsymbol{\tau} \in U(\Phi_0, \langle e, \alpha \rangle, k)$ if and only if $\boldsymbol{\tau}$ is essential to splits below Φ_0, which is equivalent to:

$$\begin{aligned}(\forall p, q <_{\mathcal{P}} (\Phi_0, \emptyset))(\forall m, m_1, m_2)[(p \Vdash \langle e, \alpha \rangle^{(C \oplus \Phi_G)}(m) \downarrow = m_1 \\ \wedge \ q \Vdash \langle e, \alpha \rangle^{(C \oplus \Phi_G)}(m) \downarrow = m_2 \\ \wedge m_1 \neq m_2) \\ \rightarrow (\exists \langle x, y, \sigma \rangle) \text{ such that } \langle x, y, \sigma \rangle \in (\Phi_p \cup \Phi_q) \setminus \Phi_0 \\ \text{yet } \sigma \text{ compatible with some } \tau \in \boldsymbol{\tau})]\end{aligned}$$

Using a similar trick as in Lemma 3.13, we rewrite this as

$$\begin{aligned}(\forall \Phi_p, \Phi_q \leq_{\mathcal{P}} \Phi_0)(\forall m, m_1, m_2)[((\exists \mathbf{X}_p)(\Phi_p, \mathbf{X}_p) \Vdash \langle e, \alpha \rangle^{(C \oplus \Phi_G)}(m) \downarrow = m_1 \\ \wedge (\exists \mathbf{X}_q)(\Phi_q, \mathbf{X}_q) \Vdash \langle e, \alpha \rangle^{(C \oplus \Phi_G)}(m) \downarrow = m_2 \\ \wedge m_1 \neq m_2) \\ \rightarrow (\exists \langle x, y, \sigma \rangle) \text{ such that } \langle x, y, \sigma \rangle \in (\Phi_p \cup \Phi_q) \setminus \Phi_0 \\ \text{yet } \sigma \text{ compatible with some } \tau \in \boldsymbol{\tau})]\end{aligned}$$

But, by Lemma 3.12, we can replace

$$(\exists \mathbf{X}_p) \left[(\Phi_p, \mathbf{X}_p) \Vdash \langle e, \alpha \rangle^{(C \oplus \Phi_G)}(m) \downarrow = m_1 \right]$$

with

$$(\exists j) \left[T(\Phi_p, \langle e, \alpha \rangle^{C \oplus \Phi_G}(m) \downarrow = m_1, j) \text{ is infinite} \right]$$

and make similar replacements for Φ_q and $\mathbf{X}_q$.

Now, "$\langle e, \alpha\rangle^{(C\oplus\Phi_G)}(m) \downarrow= m_1$" can be expressed as a $\Sigma^r_{\alpha+1}(C)$ sentence in our forcing language, uniformly in m, m_1, e, α. So then we would like to know whether a $\Pi^0_{\alpha+2}(C)$ subtree of a recursively bounded recursive tree (have to go up one level to get a Π^r sentence) is infinite, which is a $\Pi^0_{\alpha+2}(C)$ condition, uniformly in j, m, m_1, Φ_p. Hence $\boldsymbol{\tau}$ being essential to $\langle e, \alpha\rangle$-splits below Φ_0 is equivalent to a sentence with an initial block of universals, with a conditional matrix, the antecedent of which is $\Pi^0_{\alpha+2}(C)$ and the consequent uniformly recursive. Hence $\boldsymbol{\tau}$ being essential to $\langle e, \alpha\rangle$-splits below Φ_0 is a $\Pi_{\alpha+3}(C)$ property. Uniformity is clear from the analysis. □

Corollary 3.19. *For each finite use-monotone Turing functional Φ_0, each C-hyperarithmetic reduction $\langle e, \alpha\rangle$, and each natural number k, if S is not $\Delta^0_{\alpha+3}(C)$ and if there is a size k set $\boldsymbol{X}$ such that $(\Phi_0, \boldsymbol{X})$ has no $\langle e, \alpha\rangle$-splits below it, then there is such an $\boldsymbol{X}$ not containing S. Further, we can find such an $\boldsymbol{X}$ all of whose members are recursive in $C^{(\alpha+4)}$; indeed, we can construct $\boldsymbol{X}$ uniformly in e, k and $S \oplus C^{(\alpha+4)}$.*

Proof. Suppose there is a size k set $\mathbf{X}$ such that $(\Phi_0, \mathbf{X})$ has no $\langle e, \alpha\rangle$-splits. Then by Lemma 3.17, $U(\Phi_0, \langle e, \alpha\rangle, k)$ is infinite, and by Lemma 3.18 it is uniformly $\Pi^0_{\alpha+3}(C)$. By the same argument as for Corollary 3.14, this tree has a path Y recursive in $C^{(\alpha+4)}$, and yet S is not recursive in the members of the path, and consequently, S is not on the path. Furthermore, by Lemma 3.17, the reals $\mathbf{X}(Y)$ serve as the second coordinate of a condition $(\Phi_0, \mathbf{X}(Y))$, which has no $\langle e, \alpha\rangle$-splits below it.

Now, for the uniform construction we need to build a path through a $\Pi^0_{\alpha+3}(C)$ recursively bounded tree $U(\Phi_0, \langle e, \alpha\rangle, k)$ and we need to make sure S is not on

the path. $C^{(\alpha+4)}$ suffices to decide which members of the tree have an infinite part of the tree above them. Additionally, S can check whether a given node has coordinates all of which disagree with S, and as $U(\Phi_0, \langle e, \alpha\rangle, k)$ is given uniformly in the mentioned parameters, we have the desired uniformity. □

Corollary 3.20. *Suppose $p \in \mathcal{P}$ is a forcing condition such that $A \notin \boldsymbol{X}_p$, and $\langle e, \alpha\rangle$ a C-hyperarithmetic reduction. Then there is an extension of p which adds no new axioms applying to A, and does not have A in the infinite part, and diagonalizes against $\langle e, \alpha\rangle^{C\oplus\Phi_G} = D$, i.e., either forces $\langle e, \alpha\rangle^{C\oplus\Phi_G}$ is not total or there is an x such that it forces this computation on x to be convergent, yet not equal to $D(x)$.*

Proof. Fix p. The expression "$\langle e, \alpha\rangle^{C\oplus\Phi_G}$ is total" can be rendered in our forcing language as a $\Pi^r_{\alpha+2}(C)$ sentence. Then by Corollary 3.15, we can find a stronger condition q neither adding A to the infinite part nor any new axioms applying to A to the finite part, deciding this sentence. If it forces it to be false (i.e., the reduction is not total), q is the extension we want.

Otherwise, suppose q forces totality. If q has an $\langle e, \alpha\rangle^{C\oplus\Phi_G}$ split, then there is an $r \leq_{\mathcal{P}} q$ and an x such that $r \Vdash \langle e, \alpha\rangle^{C\oplus\Phi_G}(x) \downarrow \neq D(x)$. As there is an $r \leq_{\mathcal{P}} q$ forcing this sentence, then by Corollary 3.14 there is such an r which neither adds A to the infinite part nor adds any new computations applying to A to the finite part. Such an r is the extension we want.

Now suppose q has no such splits. Then, $\langle e, \alpha\rangle^{C\oplus\Phi_G}$ is determined by q. We claim we can hyperarithmetically in C determine which set q determines this to be, and so compute $\langle e, \alpha\rangle^{C\oplus\Phi_G}$ (as determined by q) hyperarithmetically in C, which implies it is not D.

To see this, recall that as $q = (\Phi_q, \mathbf{X}_q)$ has no splits, there is a finite set of reals

$\mathbf{X}$ (still not containing A) all of whose members are recursive in $C^{(\alpha+4)}$ such that $q' = (\Phi_q, \mathbf{X})$ also has no splits, by Corollary 3.19. Note q' and q are compatible.

Now, for fixed x and each y and k, search for an Φ_r such that $(\Phi_r, \mathbf{X}) \leq_{\mathcal{P}} (\Phi_q, \mathbf{X})$ and $T(\Phi_r, \langle e, \alpha \rangle^{C \oplus \Phi_G}(x) \downarrow = y, k)$ is infinite. Checking the first condition is recursive in $C^{(\alpha+4)}$, and so too is the second, as the tree is $\Pi^0_{\alpha+2}(C)$, uniformly in $\Phi_r, e, \alpha, x, y, k$; hence this search is $C^{(\alpha+4)}$-recursive. Once we find such y, k, we can output y as the value of $\langle e, \alpha \rangle^{C \oplus \Phi_G}(x)$. As q' and q are compatible and q' has no splits, y must be the value that q decides for $\langle e, \alpha \rangle^{C \oplus \Phi_G}(x)$. We know such Φ_r, y, and k exist, because if they did not, then we could force nonconvergence of the computation at x, contrary to hypothesis.

Thus, recursively in $C^{(\alpha+3)}$ we can compute what q forces $\langle e, \alpha \rangle^{C \oplus \Phi_G}$ to be, and so, as D is not hyperarithmetic in C, this computation does not compute D. $\square$

So we have shown that we can construct a generic sequence for the easy case, while still effecting the coding. We will postpone the proof of preservation of ω_1^C until we have proved results analogous to the above for the hard case.

3.2.3 The hard case

In the previous section, we proved results which we will apply to A and C, provided A is not hyperarithmetic in C, i.e., if we are in the easy case. In the hard case, these lemmas no longer apply. This is the motivation for the forcing $\mathcal{Q}_{A,B}$ (which we abbreviate as $\mathcal{Q}$ in this section), where we build into the notion of forcing the lemmas that we cannot push through for $\mathcal{P}$. Recall that, in the hard case, we choose representatives such that $A \leq_T C$. In this section when we say a condition

forces a sentence, we mean $\Vdash_{\mathcal{Q}}$.

Note that if we are successful in our goal of coding B into the join of A and Φ_G, then in the hard case we will have

$$C \oplus \Phi_G \geq_T A \oplus \Phi_G \geq_T B$$

and, consequently, $C \oplus \Phi_G \geq_T B \oplus C$. Thus, we will attempt to preserve the fact that $B \oplus C \not\geq_h D$ in the hard case. This requires us to use the language $\mathcal{L}^r_{\omega_1,\omega}(B \oplus C)$ as our language of forcing for $\mathcal{Q}$.

Definition 3.21 ($(\mathcal{Q}, B \oplus C)$-essential to $\neg\psi$ over Φ_0)**.** If Φ_0 is a finite use-monotone Turing functional which partially computes B given A, ψ is a $\Pi^r_\alpha(B \oplus C)$ sentence in $\mathcal{L}^r_{\omega_1,\omega}(B \oplus C)$, and $\boldsymbol{\tau}$ is a finite tuple of binary strings, all of the same length, we say $\boldsymbol{\tau}$ is $(\mathcal{Q}, B \oplus C)$**-essential to** $\neg\psi$ **over** Φ_0 if whenever $q = (\Phi_q, \mathbf{X}_q) \in \mathcal{Q}$ properly extends $(\Phi_0, \emptyset)$ and $q \Vdash_{\mathcal{Q}} \neg\theta_i(\vec{m})$ for one of the conjuncts $(\forall\vec{u})\theta_i(\vec{u})$ making up ψ and some tuple $\vec{m}$ of the correct length, $\Phi_q \setminus \Phi_0$ contains an axiom $\langle x, y, \sigma\rangle$ such that σ is compatible with some $\tau \in \boldsymbol{\tau}$.

We let $T_{\mathcal{Q},B\oplus C}(\Phi_0, \psi, k)$ be the tree of essential tuples of length k, as before. We will abuse notation and omit the subscripts of $T(\Phi_0, \psi, k)$. Again, $T(\Phi_0, \psi, k)$ is a subtree of a recursively bounded recursive tree. Note that $T(\Phi_0, \psi, k)$ could have branches which contain A as the limit of one of the coordinates, meaning we can't use that branch as the set of reals for a $\mathcal{Q}$ condition. Given a path Y in the tree $T(\Phi_0, \psi, k)$ (or any of our trees) and a real S we say S **does not appear on** Y if S is not the limit of any of the coordinates of Y.

Lemma 3.22. *Suppose that Φ_0 is a finite use-monotone Turing functional which partially computes B on input A, ψ is a $\Pi^r_\alpha(B \oplus C)$ sentence with $\alpha > 0$, and k is a natural number.*

(1) If there is a size k set $\boldsymbol{X}$ of reals such that $(\Phi_0, \boldsymbol{X}) \in \mathcal{Q}$ and $(\Phi_0, \boldsymbol{X})$ forces $\psi(\Phi_G)$, then $T(\Phi_0, \psi, k)$ has an infinite path on which A does not appear.

(2) If $T(\Phi_0, \psi, k)$ has an infinite path on which A does not appear, then each such path Y is naturally identified with a size k set $\boldsymbol{X}(Y)$ (not containing A) of reals such that $(\Phi_0, \boldsymbol{X}(Y)) \in \mathcal{Q}$ and $(\Phi_0, \boldsymbol{X}(Y)))$ forces $\psi(\Phi_G)$.

Proof. The proof is similar to that of Lemma 3.12. For the first claim, note that for each l, $\boldsymbol{\tau}_l = (X_1 \upharpoonright l, \ldots, X_k \upharpoonright l)$ is essential (where $(X_1, \ldots, X_k)$ enumerates such an $\mathbf{X}$), as any $q \in \mathcal{Q}$ extending $(\Phi_0, \emptyset)$ and forcing $\neg\theta_i(\vec{m})$ for some conjunct $(\forall\vec{u})\theta_i(\vec{u})$ making up ψ and some $\vec{m}$ must be incompatible with $(\Phi_0, \mathbf{X})$. Consequently, there is a new axiom in Φ_q applying to some member of $\mathbf{X}$ and so compatible with the corresponding component of $\boldsymbol{\tau}_l$. Clearly, A does not appear on the path because $(\Phi_0, \mathbf{X}) \in \mathcal{Q}$.

For the second claim, for any path Y in $T(\Phi_0, \psi, k)$ on which A does not appear, $\mathbf{X}(Y)$ (the coordinatewise union of Y) may serve as the infinite part of a $\mathcal{Q}$ condition. Then, as in Lemma 3.12, if $q <_{\mathcal{Q}} (\Phi_0, \emptyset)$ forces $\neg\theta_i(\vec{m})$ for a θ_i and an $\vec{m}$ as above, q must contain a new axiom compatible with some coordinate of each element of Y, and so must apply to some $X \in \mathbf{X}(Y)$. Consequently, such a q does not extend $(\Phi_0, \mathbf{X})$ and so $(\Phi_0, \mathbf{X}) \Vdash_Q \psi$. □

Lemma 3.23. *For each finite use-monotone Turing functional Φ_0 that correctly computes B on input A, each $\Pi^r_\alpha(B \oplus C)$ sentence ψ in $\mathcal{L}^r_{\omega_1,\omega}(B \oplus C)$ with $\alpha > 0$, and each number k, $T(\Phi_0, \psi, k)$ is $\Pi^0_\alpha(B \oplus C)$ uniformly in Φ_0, ψ, and k.*

Proof. We use the same strategy as for Lemma 3.13. Fix $\boldsymbol{\tau}$ of length k, Φ_0 and ψ.

Then $\boldsymbol{\tau}$ being essential to $\neg\psi$ over Φ_0 is equivalent to both of the following

$$(\forall\Phi_q)(\forall\vec{m})(\forall i)[((\Phi_q,\emptyset)\leq_{\mathcal{Q}}(\Phi_0,\emptyset)\ \&\ (\exists\mathbf{X})(\Phi_q,\mathbf{X})\Vdash_{\mathcal{Q}}\neg\theta_i(\vec{m}))$$
$$\rightarrow(\exists\,\langle x,y,\sigma\rangle)\text{ such that }\langle x,y,\sigma\rangle\in\Phi_q\setminus\Phi_0$$
$$\text{yet }\sigma\text{ is compatable with some }\tau\in\boldsymbol{\tau})],$$

and

$$(\forall\Phi_q)(\forall\vec{m})(\forall j)(\forall i)[((\Phi_q,\emptyset)\leq_{\mathcal{Q}}(\Phi_0,\emptyset)\ \&\ (\exists\mathbf{X},|\mathbf{X}|=j)(\Phi_q,\mathbf{X})\Vdash_{Q}\neg\theta_i(\vec{m}))$$
$$\rightarrow(\exists\,\langle x,y,\sigma\rangle)\text{ such that }\langle x,y,\sigma\rangle\in\Phi_q\setminus\Phi_0$$
$$\text{yet }\sigma\text{ is compatable with some }\tau\in\boldsymbol{\tau})].$$

(Again the $(\forall i)$ is a universal quantifier over a $(B\oplus C)$-r.e. set.)

Now suppose $\alpha=1$; then by Lemma 3.9, the antecedent in the first formulation is $A\oplus B\oplus C$-recursive uniformly in Φ_0,Φ_q,θ_i, and $\vec{m}$. As $C\geq_T A$, we can ignore A, and then $\boldsymbol{\tau}$ being essential can be rendered as a property with two universal natural number quantifiers, then a universal quantifier over a $(B\oplus C)$-r.e. set, then a $B\oplus C$-recursive matrix, uniformly in the parameters. Hence this is a $\Pi^0_1(B\oplus C)$ property.

For the inductive steps, firstly, suppose $\alpha>1$ is a successor. We now consider the second equivalent definition of $\boldsymbol{\tau}$ being essential. By Lemma 3.22, "$(\exists\mathbf{X},|\mathbf{X}|=j)(\Phi_q,\mathbf{X})\Vdash\neg\theta_i(\vec{m})$" is equivalent to $T(\Phi_q,\neg\theta_i(\vec{m}),j)$ having a path on which A does not appear. We may render this as: does there exists a length l so that for every length $l'>l$, there exists a node on the tree at level l' each coordinate of which is not an initial segment of A? Because $T(\Phi_q,\neg\theta_i(\vec{m}),j)$ is a subtree of a recursively bounded recursive tree, the second existential quantifier is actually a bounded existential quantifier. Consequently the tree having a path on which A

does not appear is $\Sigma^0_\alpha(B \oplus C)$ as, by induction, $T(\Phi_q, \neg\theta_i(\vec{m}), j)$ is $\Pi^0_{\alpha-1}(B \oplus C)$ and $A \leq_T C$.

As this is the antecedent of a conditional, we flip to $\Pi^0_\alpha(B \oplus C)$ and then our property is expressed via an initial block of universal quantifiers, and then a $\Pi^0_\alpha(B \oplus C)$ matrix, for $\Pi^0_\alpha(B \oplus C)$ as required.

Finally, suppose $\alpha > 1$ is a limit. We use the second formulation of $\boldsymbol{\tau}$ being essential. Now, by induction, $(B \oplus C)^{(\alpha)}$ can (recursively) decide whether $T(\Phi_p, \neg\theta_i(\vec{m}), j)$ has an infinite path on which A does not appear, for any conjunct $(\forall\vec{u})\theta_i(\vec{u})$ of ψ. Consequently, when α is a limit, the matrix of the second formulation for $\boldsymbol{\tau}$ being essential is recursive in $(B \oplus C)^{(\alpha)}$, and the prefix is a block of universals for $\Pi^0_\alpha(B \oplus C)$ as required. □

Corollary 3.24. *Let Φ_0 be a finite use-monotone Turing functional, ψ a $\Pi^r_\alpha(B \oplus C)$ sentence, and k a natural number. If there is a size k set $\boldsymbol{X}$ such that $A \notin X$ and $(\Phi_0, \boldsymbol{X}) \Vdash_{\mathcal{Q}} \psi$, then we can (recursively in $(B \oplus C)^{(\alpha+1)}$) find such an $\boldsymbol{X}$, uniformly in ψ, k (of course, such a set does not contain A).*

Proof. If there is such a set $\mathbf{X}$, then by Lemma 3.22, $T(\Phi_0, \psi, k)$ has an infinite path on which A does not appear. By Lemma 3.23, $T(\Phi_0, \psi, k)$ is a $\Pi^0_\alpha(B \oplus C)$ subtree of a recursively bounded recursive tree. Consequently, $(B \oplus C)^{(\alpha+1)}$ can decide membership in $T(\Phi_0, \psi, k)$, can decide whether the tree is infinite above any given node, and can decide whether any node is incompatible with A. Thus $(B \oplus C)^{(\alpha+1)}$ can construct a path Y in $T(\Phi_0, \psi, k)$ on which A does not appear, and then $(\Phi_0, \mathbf{X}(Y)) \Vdash_{\mathcal{Q}} \psi$. □

Corollary 3.25. *Let ψ be a $\Pi^r_\alpha(B \oplus C)$ sentence and k a natural number. For any condition $q \in \mathcal{Q}$ there is a stronger $r \leq_{\mathcal{Q}} q$ which we can find uniformly in Φ_q*

and $(B \oplus C)^{(\alpha+2)} \oplus \boldsymbol{X}_q$ *such that every new* $X \in \boldsymbol{X}_r$ *is recursive in* $(B \oplus C)^{(\alpha+1)}$*, and* r *decides* ψ *(and we know whether* $r \Vdash_{\mathcal{Q}} \psi$ *or* $r \Vdash_{\mathcal{Q}} \neg\psi$*).*

Proof. Fix a $q \in \mathcal{Q}$ and enumerate $\mathbf{X}_q$ as $X_1, \ldots, X_k$. There are three cases to consider: Either $T(\Phi_q, \psi, k)$ has an infinite path on which A does not appear, or $T(\Phi_q, \psi, k)$ is finite, or $T(\Phi_q, \psi, k)$ is infinite (and so has a path) but every such path contains A.

$T(\Phi_q, \psi, k)$ is a $\Pi^0_\alpha(B \oplus C)$ subtree of a recursively bounded recursive tree given uniformly in the parameters, by Lemma 3.23. Hence, the question of $T(\Phi_q, \psi, k)$ being infinite is uniformly $\Pi^0_\alpha(B \oplus C)$ and so is decidable in $(B \oplus C)^{(\alpha+1)}$. Furthermore, if it is infinite, $(B \oplus C)^{(\alpha+2)}$ can decide whether there is a path on which A does not appear, because A is recursive in C. Therefore, $(B \oplus C)^{(\alpha+2)}$ can uniformly determine which case we are in.

Case 1: If we find $T(\Phi_q, \psi, k)$ has an infinite path on which A does not appear, $(B \oplus C)^{(\alpha+1)}$ suffices to construct such a path Y as in Corollary 3.24. By Lemma 3.22, $(\Phi_q, \mathbf{X}(Y)) \Vdash_{\mathcal{Q}} \psi$ and so $r = (\Phi_q, \mathbf{X}_q \cup \mathbf{X}(Y))$ suffices and was found uniformly.

Case 2: In this case, $T(\Phi_q, \psi, k)$ is finite and so $(X_1, \ldots, X_k)$ does not form a path through it. Therefore, for some l, $\boldsymbol{\tau}_l = (X_1 \upharpoonright l, \ldots, X_k \upharpoonright l)$ is not essential to $\neg\psi$ over Φ_q. Then there is some $p <_{\mathcal{Q}} (\Phi_q, \emptyset)$, some conjunct $\theta_i(\vec{u})$ of ψ, and an $\vec{m}$ of the correct length such that $p \Vdash_{\mathcal{Q}} \neg\theta_i(\vec{m})$, yet no $\langle x, y, \sigma\rangle \in \Phi_p \setminus \Phi_q$ is compatible with any component of $\boldsymbol{\tau}_l$. As $p \Vdash \neg\theta_i(\vec{m})$, there is a j such that $T(\Phi_p, \neg\theta_i(\vec{m}), j)$ has an infinite path on which A does not appear, by Lemma 3.22.

$(B \oplus C \oplus \mathbf{X}_q)$ can recursively find all Φ_p such that $\Phi_p \leq_{\mathcal{Q}} \Phi_q$ and adds no computations applying to any $X \in \mathbf{X}_q$. Also for fixed $\theta_i, \vec{m}, j$ the question of

whether $T(\Phi_p, \neg\theta_i(\vec{m}), j)$ has an infinite path on which A does not appear can be decided by $(B \oplus C)^{(\alpha+1)}$ (as $T(\Phi_p, \neg\theta_i(\vec{m}), j)$ is at least one level of complexity lower than $T(\Phi_p, \psi, j)$). Consequently, $(B \oplus C)^{(\alpha+1)} \oplus \mathbf{X}_q$ suffices to find such a $\Phi_p, \theta_i, \vec{m}$, and j and to construct a path Y in the tree, and so $r = (\Phi_p, \mathbf{X}(Y) \cup \mathbf{X}_q)$ is the condition we want.

Case 3: In the final case $T(\Phi_q, \psi, k)$ is infinite, but every path contains A. We claim that q does not force ψ, because if $q \Vdash_{\mathcal{Q}} \psi$, then by Lemma 3.22, $\mathbf{X}_q$ forms a branch of $T(\Phi_q, \psi, k)$ not containing A. As ψ is a universal sentence and q does not force it, there is some $p \leq_{\mathcal{Q}} q$, some conjunct $\theta_i(\vec{u})$, and some tuple $\vec{m}$ such that for every $p' \leq_{\mathcal{Q}} p$, p' does not force $\theta_i(\vec{m})$. Therefore, $p \Vdash_{\mathcal{Q}} \neg\theta_i(\vec{m})$.

For such a p, there is a j such that $T(\Phi_p, \neg\theta_i(\vec{m}), j)$ has a path Y on which A does not appear. So, by Corollary 3.24, $(B \oplus C)^{(\alpha+1)}$ can find such a $\Phi_p, \theta_i, \vec{m}$ and then construct a path Y, so then, the condition we want is $r = (\Phi_p, \mathbf{X}_q \cup \mathbf{X}(Y))$. $\square$

The reader may be wondering why we are proving theorems which are the analogues of theorems which prevent us from adding A to the infinite part of a $\mathcal{P}$ condition, when all $\mathcal{Q}$ conditions have that built in. It is, in fact, the bounds on the complexity of the forcing relation that we want. These bounds will allow us to diagonalize against computing D.

Definition 3.26 ($(\mathcal{Q}, B \oplus C)$-essential to splits)**.** Let Φ_0 be a finite use-monotone Turing functional which correctly computes B on input A, and $\langle e, \alpha\rangle$ a $(B \oplus C)$-hyperarithmetic reduction. For each finite sequence $\boldsymbol{\tau}$ of binary strings of the same length, we say $\boldsymbol{\tau}$ is $(\mathcal{Q}, B \oplus C)$**-essential to** $\langle e, \alpha\rangle$**-splits below** Φ_0 if whenever $q, r \leq_{\mathcal{Q}} (\Phi_0, \emptyset)$ form an $\langle e, \alpha\rangle$-split (i.e., there is an x such that $q \Vdash_{\mathcal{Q}} \langle e, \alpha\rangle^{B \oplus C \oplus \Phi_G}(x) \downarrow = k_1, r \Vdash_{\mathcal{Q}} \langle e, \alpha\rangle^{B \oplus C \oplus \Phi_G}(x) \downarrow = k_2$, and $k_1 \neq k_2$), there is some

new axiom $\langle x, y, \sigma \rangle \in \Phi_q \cup \Phi_r$ but not in Φ_0 such that σ is compatible with some component of $\boldsymbol{\tau}$.

We let $U_{\mathcal{Q},B\oplus C}(\Phi_0, \langle e, \alpha \rangle, k)$ denote the set of length k vectors $\boldsymbol{\tau}$ which are $(\mathcal{Q}, B \oplus C)$-essential to $\langle e, \alpha \rangle$-splits below Φ_0. As with $U_{\mathcal{P},C}(\Phi_0, \langle e, \alpha \rangle, k)$, this set can be considered as a subtree of a recursively bounded recursive tree. We drop the $\mathcal{Q}$ and $B \oplus C$ decorations wherever possible.

Lemma 3.27. *Let Φ_0 be a Turing functional which partially computes B on input A, $\langle e, \alpha \rangle$ a $B \oplus C$-hyperarithmetic reduction, and k a natural number.*

- *If $\boldsymbol{X}$ is a size k set such that $(\Phi_0, \boldsymbol{X}) \in \mathcal{Q}$, and no pair $q, r \leq_{\mathcal{Q}} (\Phi_0, \boldsymbol{X})$ forms an $\langle e, \alpha \rangle$-split, then $U(\Phi_0, \langle e, \alpha \rangle, k)$ has an infinite path on which A does not appear.*
- *If $U(\Phi_0, \langle e, \alpha \rangle, k)$ has an infinite path on which A does not appear, then each such path Y is identified with a size k set $\boldsymbol{X}(Y)$ such that $(\Phi_0, \boldsymbol{X}(Y)) \in \mathcal{Q}$ has no $\langle e, \alpha \rangle$-splits below it.*

Proof. The strategy is similar to that of Lemma 3.17, with the same modifications needed to turn a proof of Lemma 3.12 into one of Lemma 3.22. □

Lemma 3.28. *If Φ_0 is a Turing functional which partially computes B on input A, $\langle e, \alpha \rangle$ a $(B \oplus C)$-hyperarithmetic reduction, and k a natural number, then $U(\Phi_0, \langle e, \alpha \rangle, k)$ is $\Pi_{\alpha+3}(B \oplus C)$, uniformly in the defining parameters. Furthermore, if there exists a size k set $\boldsymbol{X}$, such that $(\Phi_0, \boldsymbol{X}) \in \mathcal{Q}$ has no $\langle e, \alpha \rangle$-splits below it, then we can find such an $\boldsymbol{X}$ not containing A all of whose members are recursive in $(B \oplus C)^{(\alpha+4)}$ (it is also uniform, although this is not needed).*

Proof. The strategy is similar to that of Lemmas 3.18 and 3.19. Fix $\Phi_0, k, \langle e, \alpha \rangle$, and $\boldsymbol{\tau}$. Then $\boldsymbol{\tau} \in U(\Phi_0, \langle e, \alpha \rangle, k)$ iff $(\forall \Phi_q, \Phi_r \leq_{\mathcal{Q}} \Phi_0)(\forall m, m_1 \neq m_2)$

$$\begin{aligned}
&(\exists \mathbf{X}_q, A \notin \mathbf{X}_q)((\Phi_q, \mathbf{X}_q) \Vdash \langle e, \alpha \rangle^{(B \oplus C \oplus \Phi_G)}(m) \downarrow = m_1) \\
\wedge &(\exists \mathbf{X}_r, A \notin \mathbf{X}_r)((\Phi_r, \mathbf{X}_r) \Vdash \langle e, \alpha \rangle^{(B \oplus C \oplus \Phi_G)}(m) \downarrow = m_2) \\
\rightarrow &(\exists \langle x, y, \sigma \rangle) \text{ such that } \langle x, y, \sigma \rangle \in (\Phi_q \cup \Phi_r) \setminus \Phi_0 \\
&\text{yet } \sigma \text{ is compatible with some } \tau \in \boldsymbol{\tau})
\end{aligned}$$

By Lemma 3.27, we can replace the existential quantifiers over reals by

$(\exists j) U(\Phi_q, \langle e, \alpha \rangle^{(B \oplus C \oplus \Phi_G)}(m) \downarrow = m_1, j)$ has a path on which A does not appear.

(making the analogous changes for Φ_r). Now, the formula "$\langle e, \alpha \rangle^{(B \oplus C \oplus \Phi_G)}(m) \downarrow = m_1$" can be rendered as a $\Sigma^r_{\alpha+1}(B \oplus C)$ formula in our forcing language, so increasing its level by one to get a $\Pi^r(B \oplus C)$ formula, and applying Lemma 3.23, the existential quantifier over reals is equivalent to a (uniformly given) $\Pi^0_{\alpha+2}(B \oplus C)$ tree having a path on which A does not appear.

We can decide whether the tree has such a path by answering the question: Does there exist a j and a node on a uniformly given $\Pi^0_{\alpha+2}(B \oplus C)$ subtree of a recursively bounded tree, the coordinates of which all disagree with A, above which the tree is infinite? This is a uniformly $\Sigma^0_{\alpha+3}(B \oplus C)$ question.

This means we can express "$\boldsymbol{\tau}$ is essential" as a formula with a prefix of universal quantifiers, then a uniformly given $\Pi^0_{\alpha+3}(B \oplus C)$ matrix, and so this property is $\Pi^0_{\alpha+3}(B \oplus C)$, as required.

So, to find a "simple" $\mathbf{X}$, provided there is one, we need to be able to construct the tree $U(\Phi_0, \langle e, \alpha \rangle, k)$ and find paths on it on which A does not appear. As $U(\Phi_0, \langle e, \alpha \rangle, k)$ is a $\Pi^0_{\alpha+3}(B \oplus C)$ subtree of a recursively bounded recursive tree,

$(B \oplus C)^{(\alpha+4)}$ can decide whether a given $\boldsymbol{\tau}$ is on the tree and, if so, whether the tree is infinite above that node. It can also decide whether the node has a coordinate which is an initial segment of A, and so can construct the desired path, provided one exists. □

Corollary 3.29. *Let $q \in \mathcal{Q}$ be a forcing condition and $\langle e, \alpha \rangle$ a $(B \oplus C)$-hyperarithmetic reduction. Then we can find an extension $r \in \mathcal{Q}$ of q which diagonalizes against $\langle e, \alpha \rangle^{B \oplus C \oplus \Phi_G} = D$, i.e., p either forces $\langle e, \alpha \rangle^{B \oplus C \oplus \Phi_G}$ is not total, or, there is an x such that it forces this computation on x to be convergent, yet not equal to $D(x)$.*

Proof. The idea is the same as in Corollary 3.20. We assume there is a condition q which forces totality and has no $\langle e, \alpha \rangle$-splits. We then move to the related condition q' which has the same finite part as q, but the real parts are all recursive in $(B \oplus C)^{(\alpha+4)}$ as guaranteed by Lemma 3.28. We then search for extensions of q' forcing $\langle e, \alpha \rangle^{B \oplus C \oplus \Phi_G}(x) \downarrow = y$ and output y if we find it. We only need to search for conditions with new reals recursive in $(B \oplus C)^{(\alpha+1)}$ by Corollary 3.25. This makes the search hyperarithmetic in $B \oplus C$, and so, if it computes D, then $B \oplus C$ computes D, contrary to our hypothesis. □

3.2.4 Preserving the computable ordinals

A crucial step when forcing in the hyperarithmetic setting is to show that we can also preserve ω_1^{CK} relativised to C in the easy case, and $B \oplus C$ in the hard case. We use a method suggested by Slaman: forcing over nonstandard models of ZFC.

Definition 3.30. Let $\mathcal{M}$ be an ω-model of ZFC. Then $\mathcal{P}_{\mathcal{M}}$ is the collection of all Kumabe-Slaman conditions $p = (\Phi_p, \mathbf{X}_p) \in \mathcal{P}$ such that every real $X \in \mathbf{X}_p$ is in

$\mathcal{M}$. Because $\mathcal{M}$ is an ω-model this definition is meaningful. Similarly, $\mathcal{Q}_\mathcal{M}$ is the restriction of the forcing $\mathcal{Q} = \mathcal{Q}_{A,B}$ to $\mathcal{M}$. Extension is defined as for $\mathcal{P}$ and $\mathcal{Q}$ respectively. Note that $\mathcal{P}_\mathcal{M} \in \mathcal{M}$ for any such $\mathcal{M}$, and $\mathcal{Q}_\mathcal{M} \in \mathcal{M}$ if A and B are also members of $\mathcal{M}$.

Notation. Given a Kumabe-Slaman condition $p = (\Phi_p, \mathbf{X}_p) \in \mathcal{P}$, we call $(\Phi_p, \mathbf{X}_p \cap (2^\omega)^\mathcal{M})$ the restriction of p to $\mathcal{M}$ and denote it $p \restriction_\mathcal{M}$.

Easy case

Suppose $C \in \mathbf{c}$ corresponds to an easy case, and $\mathcal{M}$ is a countable ω-model of ZFC that omits ω_1^C and A but contains C. The existence of such a model is shown in [9].

Lemma 3.31. *If $(p_i)_{i=0}^\infty$ is an $\mathcal{M}$-generic sequence (i.e. the sequence meets all of the dense subsets of $\mathcal{P}_\mathcal{M}$ that appear in $\mathcal{M}$), then Φ_G preserves ω_1^C, that is to say, $\omega_1^{C \oplus \Phi_G} = \omega_1^C$.*

Proof. Suppose $<$ is a well-order recursive in $C \oplus \Phi_G$. As $\mathcal{M}$ is an ω-model and the ordinals of $\mathcal{M}[\Phi_G]$ are the same as that of $\mathcal{M}$, it follows that $\mathcal{M}[\Phi_G]$ is an ω-model containing $C \oplus \Phi_G$ and is closed under Turing reduction. Consequently, $\mathcal{M}[\Phi_G]$ contains an isomorphic copy of $<$, and as $<$ is externally well-founded, it is well-founded in $\mathcal{M}[\Phi_G]$.

Hence $\mathcal{M}[\Phi_G]$ thinks $<$ is isomorphic to an ordinal α, and as $<$ is externally well-founded, so too is α. However, the only externally well-founded ordinals in $\mathcal{M}[\Phi_G]$ are those that are also externally well-founded ordinals in $\mathcal{M}$, and by hypothesis, those are precisely the ordinals less than ω_1^C. Therefore, $<$ has height less than ω_1^C and so $\omega_1^{C \oplus \Phi_G} = \omega_1^C$ as required. $\square$

Lemma 3.32. *Suppose $\mathcal{D} \subseteq \mathcal{P}_{\mathcal{M}}$ is dense and also an element of $\mathcal{M}$. Further suppose $p = (\Phi_p, \boldsymbol{X}_p) \in \mathcal{M}$ is a forcing condition. Then we can find an extension $q \in \mathcal{P}_{\mathcal{M}}$ of p meeting $\mathcal{D}$ such that q adds no new computations applying to A, and A is not in the infinite part of q (as $A \notin \mathcal{M}$).*

Proof. Suppose we can't meet $\mathcal{D}$ below p without adding new computations to A. We will show that $\mathcal{D}$ is not dense. Let $\boldsymbol{\tau}$ be a finite sequence of finite binary strings all of the same length. We say $\boldsymbol{\tau}$ is essential to meeting $\mathcal{D}$ over Φ_p if for every condition $r = (\Phi_r, \mathbf{X}_r) \in \mathcal{D}$ with $r < (\Phi_p, \emptyset)$, there is an axiom $\langle x, y, \sigma \rangle \in \Phi_r \setminus \Phi_p$ such that σ is compatible with some component of $\boldsymbol{\tau}$.

Let $V(\Phi_p, \mathcal{D}, k)$ denote the set of $\boldsymbol{\tau}$ of length k which are essential to meeting $\mathcal{D}$ over Φ_p and note $V(\Phi_p, \mathcal{D}, k) \in \mathcal{M}$ for all k. Let $k = |\mathbf{X}_p|$, and enumerate $\mathbf{X}_p = (X_1, \ldots, X_k)$. We claim $\boldsymbol{\tau}_l = (X_1 \upharpoonright l, \ldots, X_k \upharpoonright l, A \upharpoonright l) \in V(\Phi_p, \mathcal{D}, k+1)$ for each l. To see this, suppose $r = (\Phi_r, \mathbf{X}_r) \in \mathcal{P}_{\mathcal{M}}$ properly extends $(\Phi_p, \emptyset)$ and meets D. If r has no new axioms applying to any of $X_1, \ldots, X_k, A$, then r extends p, contradicting our hypothesis that we cannot meet D below p without adding axioms about A. Hence, such an r has a new axiom $\langle x, y, \sigma \rangle$ applying to one of $X_1, \ldots, X_k, A$, and so σ is compatible with the corresponding component of $\boldsymbol{\tau}_l$, showing $\boldsymbol{\tau}_l$ is essential.

Consequently, $V(\Phi_p, \mathcal{D}, k+1)$ is an infinite finitely-branching tree that is an element of $\mathcal{M}$. As $\mathcal{M}$ is an ω-model, it sees that $V(\Phi_p, \mathcal{D}, k+1)$ is infinite and finitely-branching, so by König's Lemma inside $\mathcal{M}$, $V(\Phi_p, \mathcal{D}, k+1)$ has a branch Y in $\mathcal{M}$. This branch corresponds to reals $\mathbf{X}(Y)$ which are also in $\mathcal{M}$ and so $q = (\Phi_p, \mathbf{X}_p \cup \mathbf{X}(Y)) \in \mathcal{P}_{\mathcal{M}}$ and q extends p.

As $\mathcal{D}$ is dense, there is some extension $r = (\Phi_r, \mathbf{X}_r)$ of q in $\mathcal{D}$. Such an r extends

$(\Phi_p, \emptyset)$, so by the definition of essentiality, there is some axiom $\langle x, y, \sigma \rangle \in \Phi_r \setminus \Phi_p$ with σ compatible with some component of each element of Y. Therefore, σ is an initial segment of some $X \in \mathbf{X}(Y)$, but the axiom $\langle x, y, \sigma \rangle$ can only be in an extension of $q = (\Phi_p, \mathbf{X}_p \cup \mathbf{X}(Y))$ if it was already in Φ_p, a contradiction. Therefore, q has no extensions in $\mathcal{D}$ and so $\mathcal{D}$ is not dense. □

Hard case

If $A \leq_h C$, then we cannot omit A from ω-models of ZFC containing C. Consequently, we cannot rely on the fact that $A \notin \mathcal{M}$ to prevent us from adding A to the infinite part of a condition. Indeed, if $\mathcal{M}$ contains A, then $\mathcal{D} = \{p \in \mathcal{P} : A \in \mathbf{X}_p\}$ is a dense subset of $\mathcal{P}_\mathcal{M}$ which is also a member of $\mathcal{M}$. Consequently, we have to use a different forcing.

Of course, given what has come before, we want to use $\mathcal{Q}_\mathcal{M}$. Any ω-model which contains C must contain A in the hard case. Consequently, $\mathcal{Q}_\mathcal{M}$ is an element of such a model iff B is. Therefore, in the hard case, we pick an ω-model $\mathcal{M}$ which contains C and B yet omits $\omega_1^{B \oplus C}$; again this follows from [9].

Once one has such a model, the argument for any set forcing preserving the least omitted ordinal (in this case $\omega_1^{B \oplus C}$) is the same as above. By definition of the forcing, we need not worry about interfering with our coding procedure.

3.2.5 The compatibility lemma

For a condition $p = (\Phi_p, \mathbf{X}_p)$ in either of the restricted forcings $\mathcal{P}_\mathcal{M}$ and $\mathcal{Q}_\mathcal{M}$, it is still a fact that no extension of p (in the relevant forcing) can add new axioms

about members of $\mathbf{X}_p$. However, as the class of reals from which we can pick members of $\mathbf{X}_p$ from is highly restricted, we want to know we can meet dense sets without adding axioms applying to reals not appearing in the models (to ensure compatibility with the unrestricted forcings).

Lemma 3.33. *Suppose C is an easy case. Then for every countable ω-model $\mathcal{M}$ of ZFC omitting ω_1^C and A yet containing C, every finite set of reals $\boldsymbol{X}$, every condition $p = (\Phi_p, \boldsymbol{X}_p) \in \mathcal{P}_\mathcal{M}$, and every dense $\mathcal{D} \subseteq \mathcal{P}_\mathcal{M}$ which is also an element of $\mathcal{M}$, there is an extension $r \in \mathcal{P}_\mathcal{M}$ of p meeting $\mathcal{D}$ which adds no new computations to any member of $\boldsymbol{X}$.*

On the other hand, if C is a hard case, then for every countable ω-model $\mathcal{M}$ of ZFC omitting $\omega_1^{B\oplus C}$ yet containing B and C (and, consequently, A), every finite set of reals $\boldsymbol{X}$ not containing A, every condition $q \in \mathcal{Q}_\mathcal{M}$, and every dense $\mathcal{D} \subseteq \mathcal{Q}_\mathcal{M}$ which is also an element of $\mathcal{M}$, there is an extension $r \in \mathcal{Q}_\mathcal{M}$ of q meeting $\mathcal{D}$ which adds no new computations to any member of $\boldsymbol{X}$.

Proof. For the first claim, fix such an $\mathcal{M}$, an $\mathbf{X}$, a $p \in \mathcal{P}_\mathcal{M}$, and a $\mathcal{D} \in \mathcal{M}$. If any $X \in \mathbf{X}$ is also an element of $\mathcal{M}$, then the condition $p' = (\Phi_p, \mathbf{X}_p \cup (\mathbf{X} \cap \mathcal{M})) \in \mathcal{P}_\mathcal{M}$ extends p and can add no new computations. By the density of $\mathcal{D}$, we can meet $\mathcal{D}$ below p' adding no computations to any $X \in \mathbf{X} \cap \mathcal{M}$, so we assume that $\mathbf{X}$ is disjoint from $\mathcal{M}$.

Now suppose that every extension of p in $\mathcal{D}$ adds a new axiom applying to some $X \in \mathbf{X}$. As in Lemma 3.32, consider the tree $V(\Phi_p, \mathcal{D}, |\mathbf{X}|)$ of essential $\boldsymbol{\tau}$ of length $|\mathbf{X}|$. By assumption, $\boldsymbol{\tau}_l = (X_1 \upharpoonright l, \ldots, X_n \upharpoonright l)$ is essential to meeting $\mathcal{D}$ below p (where $\mathbf{X} = \{X_1, \ldots X_n\}$). Consequently, $V(\Phi_p, \mathcal{D}, |\mathbf{X}|)$ is a finitely branching infinite tree in $\mathcal{M}$. Hence, it has a path Y in $\mathcal{M}$ corresponding to reals $\mathbf{X}(Y)$ also in $\mathcal{M}$.

We claim $(\Phi_p, \mathbf{X}(Y))$ has no extensions in $\mathcal{D}$. If it has, pick one r. Such an r has a new axiom compatible with some component of each member of Y. Consequently, r has a new axiom applying to some $X \in \mathbf{X}(Y)$ and so r can't extend $(\Phi_p, \mathbf{X}(Y))$, a contradiction.

The proof for the second claim in the lemma is similar: as we are assuming $\mathbf{X}$ does not contain A, we can still use the same trick as before to assume $\mathbf{X}$ is disjoint from $\mathcal{M}$. The proof from there is the same. □

3.2.6 Bringing it all together

Fix representatives $A \in \mathbf{a}$ and $B \in \mathbf{b}$. Then for each $\mathbf{c}_i, \mathbf{d}_i$ pair, pick representatives C_i and D_i, with the requirement that if i is a hard case, then $A \leq_T C_i$. For each easy case i, fix a countable ω-model $\mathcal{M}_i$ of ZFC containing C_i yet omitting $\omega_1^{C_i}$ and A, and for each hard case j, fix a countable ω-model $\mathcal{M}_j$ of ZFC containing B and C_i yet omitting $\omega_1^{B \oplus C_i}$.

Now we need to show that we can construct a generic for all the forcings simultaneously. We can't construct a single generic sequence, because we may add a real X to the infinite part at some point, where that X is not a member of some $\mathcal{M}_i$. To work around this, we use the Compatibility Lemma.

We will have one master sequence $\{(\Phi_n, \mathbf{X}_n)\}_{n=0}^{\infty}$ which will live in $\mathcal{P}$, the unrestricted forcing. We will maintain that $A \notin \mathbf{X}_n$, Φ_n partially computes B on input A, and $p_{n+1} \leq_{\mathcal{P}} p_n$. These three conditions will guarantee that our sequence is also a descending sequence of $\mathcal{Q}$ conditions. The master sequence will be used to decide all the sentences of the languages $\mathcal{L}^r_{\omega_1,\omega}(C_i)$ and $\mathcal{L}^r_{\omega_1,\omega}(B \oplus C_j)$ for easy cases i and hard cases j. We will also diagonalize against computing D_i. Corollaries

3.15, 3.25, 3.20 and 3.29 together imply we can build such a sequence.

To meet the dense sets in one of our ω-models $\mathcal{M}_i$, we consider the sequence of restricted conditions $(\Phi_n, \mathbf{X}_n) \upharpoonright \mathcal{M}_i$. Certainly at some stage of the construction, we can extend the condition $(\Phi_n, \mathbf{X}_n) \upharpoonright \mathcal{M}_i$ in the corresponding forcing living in $\mathcal{M}_i$ to meet a dense set. However, we can do better.

By Lemma 3.33, we can extend $(\Phi_n, \mathbf{X}_n) \upharpoonright \mathcal{M}_i$ in $\mathcal{M}_i$ to meet $\mathcal{D}$, without adding any axioms applying to A or any member of $\mathbf{X}_n$, and without adding A to the infinite part. Call such an extension $q = (\Phi_q, \mathbf{X}_q)$. Then we would define the next member of the master sequence to be $p_{n+1} = (\Phi_q, \mathbf{X}_q \cup \mathbf{X}_{p_n})$. By hypothesis, $p_{n+1} \leq_{\mathcal{P}} p_n$, p_{n+1} partially computes B given A, and does not have A in the infinite part. Thus, we can continue coding, and have maintained our invariants.

Finally we need to avoid the ideals below each $\mathbf{e}_i$. There is an indirect argument that does this for free: The collection of sets hyperarithmetic in any of the $\mathbf{e}_i$s is countable. As there are continuum many generics (even with our coding procedure), then there must be a generic satisfying the above that is not hyperarithmetic in $\mathbf{e}_i$ for each i. Alternatively, we could directly diagonalize against Φ_G being equal to any set $Y \leq_h \mathbf{e}_i$, we will sketch this argument as it will be useful later.

Without loss of generality we may assume Y is an infinite use-monotone Turing functional that partially computes B on input A (as Φ_G satisfies these properties), which is compatible with our current condition $(\Phi_n, \mathbf{X}_n)$. Suppose $Y \setminus \Phi_n$ has an axiom not applying to A. Then we may add a real X to which the axiom does apply to $\mathbf{X}_n$, and that axiom can't enter at a later stage, hence no generic compatible with current sequence is equal to Y. If there is no such axiom, then all of Ys axioms (not already in Φ_n) apply to A. We may assume that Y has axioms

for all inputs x, as Φ_G will have such axioms. Pick an x such that Φ_n has no axiom $\langle x, y, \alpha\rangle$ applying to A. As Y is a use-monotone Turing functional, there is a unique y and $\alpha \subset A$ such that $\langle x, y, \alpha\rangle \in A$. Then, to diagonalize, merely pick a different initial segment α' of A such that $\langle x, y, \alpha'\rangle$ is not prevented from entering $(\Phi_n, \mathbf{X}_n)$ and add that to Φ_n.

Consequently, our master sequence $(\Phi_n, \mathbf{X}_n)$ decides each sentence of each of our languages $\mathcal{L}^r_{\omega_1,\omega}(C_i)$ and $\mathcal{L}^r_{\omega_1,\omega}(B \oplus C_j)$, diagonalizes against every C_i or $B \oplus C_j$-hyperarithmetic reduction, depending on the respective case, and Φ_G is not in any of the ideals below an $\mathbf{e}_i$. The restricted sequence $(\Phi_n, \mathbf{X}_n) \upharpoonright \mathcal{M}_i$ is $\mathcal{M}_i$ generic and corresponds to the same generic object as for the master sequence. Hence Φ_G preserves each $\omega_1^{C_i}$ and $\omega_1^{B \oplus C_j}$, which completes the proof of Theorem 3.3.

3.3 Greatest element preserving extensions

It is of general interest to try to compute the complexity of the generic object (or more precisely, to determine whether we can produce generics that are sufficiently simple); paying careful attention to the construction can yield new results.

We prove bounded versions of the results of Sections 3.1 and 3.2. These can be interpreted as solving the positive half of the extension of embeddings problem for $\mathcal{D}_h(\leq_h \mathcal{O})$ because every finite almost end extension $\mathscr{V}$ of a finite USL$^\top$ $\mathscr{U}$ is a subUSL$^\top$ of a simple almost end extension of a finite $\top$-preserving free extension of $\mathscr{U}$. The proof of this is Theorem A.2 in Appendix A.

3.3.1 Free extensions

Let $\mathscr{U}$ be a finite $\mathrm{USL}^\top$. The notion of free extensions for this structure is slightly more complicated because we need to preserve the greatest element. Consequently, instead of taking the poset product of $\mathscr{U}$ and the poset of finite subsets of a set X, we take the poset product of $\mathscr{U} \setminus \{\top\}$ with the poset of finite subsets of X, and then declare $\top$ to be bigger than everything in this product. The details can be found in Appendix A.

The bounded analogue of Theorem 3.2 is as follows:

Theorem 3.34. *If $\mathscr{U}$ is a finite $\mathrm{USL}^\top$ and $\mathscr{V}$ is the $\top$-preserving free extension of $\mathscr{U}$ by finitely many free generators, then every embedding of $\mathscr{U}$ into $\mathcal{D}_h(\leq_h \mathcal{O})$ (i.e. every injective map from $\mathscr{U}$ into $\mathcal{D}_h(\leq_h \mathcal{O})$ that preserves $\bot, \top$, and $\sqcup$) extends to an embedding of $\mathscr{V}$.*

The idea of the proof is the same as that of Theorem 3.2: we want to pick a Cohen generic G and use the columns as generators. We need our G to be sufficiently generic so that the columns are independent (in an appropriate sense), but also to be a member of $\mathcal{D}_h(\leq_h \mathcal{O})$.

Lemma 3.35. *Let $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ be hyperarithmetic degrees all strictly hyperarithmetic in $\mathcal{O}$. Then there is a Cohen generic G that is strictly hyperarithmetic in $\mathcal{O}$ and that is simultaneously generic relative to each $\boldsymbol{a}_i$.*

By Corollary 2.23, $\mathcal{O}^{\mathbf{a}_i} \equiv_h \mathcal{O}$ for $i = 1, \ldots, n$. Consequently, the relativisation of the fact that there is a Cohen generic (relative to $\emptyset$) strictly hyperarithmetic in $\mathcal{O}$ is that there is a Cohen generic (relative to $\mathbf{a}_i$) strictly hyperarithmetic in $\mathcal{O}^{\mathbf{a}_i} \equiv_h \mathcal{O}$ given that $\mathbf{a}_i <_h \mathcal{O}$. There is no extra trouble in constructing

a simultaneous generic for finitely many parameters, because you can extend a condition to decide something for $\mathbf{a}_1$ without, say, interfering with compatibility for $\mathbf{a}_2$.

Fix $\mathscr{U}, \mathscr{V}$, and f as in Theorem 3.34. Using Lemma 3.35 pick a Cohen generic strictly hyperarithmetic in $\mathcal{O}$ that meets every dense subset of Cohen forcing hyperarithmetic in $f(u)$ for any $u \in \mathscr{U} \setminus \{\top\}$. Extend f by fixing a free generating set $\{g_1, \ldots, g_n\}$ of $\mathscr{V}$ over $\mathscr{U}$, mapping g_i to the degree of $G^{[i]}$, and extending to the rest of $\mathscr{V}$ using the join structure. It can be shown that this map is a $\mathrm{USL}^\top$ embedding of $\mathscr{V}$ as required in a manner analogous to that used in the proof of Theorem 3.2.

3.3.2 Simple almost end extensions

The definition of simple extension for $\mathrm{USL}^\top$s is the same as for USLs: generated by a single new element. The notion of end extension does not make sense though because $\top$ must be preserved. The next best thing is an almost end extension which is an extension that puts no new element below any old element except $\top$. We make the same reduction from the full simple almost end extension case to a subcase as in Section 3.2. The proof of the sufficiency is Theorem A.4 in Appendix A.2.

Theorem 3.36. *Let $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}_i, \boldsymbol{d}_i$ be degrees in $\mathcal{D}_h(\leq_h \mathcal{O})$ for $i = 1, \ldots, n$ satisfying*

$$\bigwedge_{i=1}^{n} \boldsymbol{d}_i \not\leq_h \boldsymbol{c}_i \,\&\, (\boldsymbol{a} \not\leq_h \boldsymbol{c}_i \text{ or } \boldsymbol{d}_i \not\leq_h \boldsymbol{b} \oplus \boldsymbol{c}_i).$$

Then for every $\boldsymbol{e}_j \in \mathcal{D}_h(< \mathcal{O})$ where $j = 1, \ldots, m$ there is a $g < \mathcal{O}$ such that

$$\boldsymbol{b} \leq \boldsymbol{a} \oplus g \,\&\, \bigwedge_{i=1}^{n} \boldsymbol{d}_i \not\leq_h \boldsymbol{c}_i \oplus g \,\&\, \bigwedge_{j=1}^{m} g \not\leq_h \boldsymbol{e}_j.$$

The proof of this theorem involves, essentially, checking that the proof of Theorem 3.3 can be made hyperarithmetic in $\mathcal{O}$ provided the parameters are strictly hyperarithmetic in $\mathcal{O}$. We outline that process here.

Suppose the antecedent in the statement of Theorem 3.36 and let $A \in \mathbf{a}, B \in \mathbf{b}, C_i \in \mathbf{c}_i, D_i \in \mathbf{d}_i$ and $E_j \in \mathbf{e}_j$ be representatives of the degrees. Our previous construction used Kumabe-Slaman forcing to produce a sequence of forcing conditions $\{(\Phi_i, \mathbf{X}_i) : i \in \omega\}$ where each Φ_i is a finite use monotone Turing functional, and each $\mathbf{X}_i$ is a finite set of reals. We want to check that $\mathcal{O}$ can produce a generic of the correct kind.

The construction is complicated by the disjunction in the antecedent, i.e., whether we are in the easy case or the hard case. As there are only finitely many pairs, we can hard code into our algorithm which cases are easy and which are hard, so $\mathcal{O}$ need not be able to determine this uniformly.

The coding procedure for cupping $\mathbf{a}$ above $\mathbf{b}$ is to add axioms (x, y, α) to the generic object such that $\alpha \subset A$ and $B(x) = y$. Hence, the coding procedure is uniform and recursive in $A \oplus B$, and, as such, is hyperarithmetic in $\mathcal{O}$.

We need $\mathcal{O}$ to be able to manipulate forcing conditions and sentences in various forcing languages. Recall that to the easy case we associate the usual Kumabe-Slaman forcing and a forcing language $\mathcal{L}^r_{\omega_1,\omega}(C_i)$, and to the hard case a restricted version of the forcing and the language $\mathcal{L}^r_{\omega_1,\omega}(B \oplus C_i)$. The restricted version of the forcing consists of all the conditions which do not explicitly get the coding procedure wrong. Importantly, the restriction is hyperarithmetic in $A \oplus B$ and so hyperarithmetic in $\mathcal{O}$. Additionally, the languages are recursive in $\omega_1^{C_i}$ and $\omega_1^{B \oplus C_i}$, respectively. As C_i does not compute D_i in the easy case and $B \oplus C_i$ does

not compute D_i in the hard case, neither of these are equal to $\mathcal{O}$, hence, their hyperjumps are hyperarithmetically equivalent to $\mathcal{O}$ as in Corollary 2.23, and so the languages of forcing are hyperarithmetic in $\mathcal{O}$.

So, now we need to show that $\mathcal{O}$ can construct a generic sequence of conditions while maintaining the coding. We will construct one sequence $\{(\Phi_{p_n}, \mathbf{X}_{p_n})\}_{i=0}^{\infty}$ of Kumabe-Slaman conditions called the **master sequence**. To keep the complexity of the construction down, we need to make sure that the reals we add to the infinite part of a condition $\mathbf{X}$ are simple. This requires a little extra technical work; roughly speaking, say we are trying to meet a dense set corresponding to some fact we wish to force about $\Phi_G \oplus C_i$ and we have a condition $(\Phi_{p_n}, \mathbf{X}_{p_n})$ corresponding to what we have done so far. We will temporarily "forget" reals $X \in \mathbf{X}_{p_n}$ which are too complicated, to get a modified condition $(\Phi_{p_n}, \mathbf{X}'_{p_n})$. We will find a simple extension of this condition which forces whichever fact we are trying to force, and then we will reinsert the "forgotten" reals. There is an obvious worry that this new condition need not refine $(\Phi_{p_n}, \mathbf{X}_{p_n})$ as we may have added computations applying to the forgotten reals, so we need to show that we can find an extension of $(\Phi_{p_n}, \mathbf{X}'_{p_n})$ which doesn't add new axioms to Φ_{p_n} which apply to the reals we have forgotten. This procedure is analogous to the compatibility lemma.

We start with the condition $p_0 = (\emptyset, \emptyset)$. We must decide each sentence of our forcing languages. What we do depends on whether we are in the easy or hard case. Note that, given a real S strictly hyperarithmetic in $\mathcal{O}$ and a finite set $\mathbf{X}$ of reals each strictly hyperarithmetic in $\mathcal{O}$, that $\mathcal{O}$ can uniformly determine which of the $X \in \mathbf{X}$ are hyperarithmetic in S. Hence, the "forgetting" procedure mentioned above can be made effective in $\mathcal{O}$. We will preserve throughout that for each condition p_n of our master sequence, each $X \in \mathbf{X}_{p_n}$ will be strictly hyperarithmetic

in $\mathcal{O}$ (although their join may not be).

In the easy case Corollary 3.15 says that given a sentence φ and a condition $(\Phi_p, \mathbf{X}_p)$, we can find an extension deciding φ, without messing up the coding, uniformly in $A \oplus C_i^{(\alpha+1)} \oplus \mathbf{X}_p$ (where α is an ordinal less than $\omega_1^{C_i} = \omega_1^{\mathrm{CK}}$ which measures the complexity of φ). So suppose we have a condition $p_n = (\Phi_{p_n}, \mathbf{X}_{p_n})$ such that Φ_n has coded correctly so far and $\mathbf{X}_{p_n}$ does not contain A. Let $p' = (\Phi_{p_n}, \mathbf{X}'_{p_n})$ where $\mathbf{X}'_{p_n}$ is the intersection of $\mathbf{X}_{p_n}$ with the set of reals which are hyperarithmetic in C_i. We can extend Corollary 3.15 so that we can pick our extension to not add any axioms to any $X \in \mathbf{X}_{p_n} \setminus \mathbf{X}'_{p_n}$ very easily. The proof already does this for an arbitrary S which is not $\Delta_0^{(\alpha+1)}$. The proof for finitely many such S goes through in the same way, and as each $X \in \mathbf{X}_{p_n} \setminus \mathbf{X}'_{p_n}$ is not hyperarithmetic in C_i we can apply this result to those reals. Thus, we produce an extension q' of p' which decides the sentence, such that $q = (\Phi_{q'}, \mathbf{X}_{q'} \cup \mathbf{X}_{p_n})$ extends p_n, does not interfere with the coding, and each $X \in \mathbf{X}_{q'}$ is hyperarithmetic in C. We can then define $p_{n+1} = q$.

In the hard case we rely on Corollary 3.25 (with similar modifications as in the easy case) to forget the reals not hyperarithmetic in $B \oplus C_i$, and find an extension deciding a sentence which is compatible with the starting condition.

We also need to diagonalize against cupping C_i above D_i (in the hard case we diagonalize against cupping $B \oplus C_i$ above D_i). The relevant results are Corollaries 3.19 and 3.28, respectively. Although it is not observed directly in the statements of these Corollaries, it is clear from the proofs that the diagonalizing extension can be found uniformly in (some jump of) the previous condition (where the number of jumps needed is tied nicely to the complexity of the reduction against which we are diagonalizing). Consequently, our trick of temporarily forgetting reals which aren't

hyperarithmetic in C_i (or $B \oplus C_i$) allows us to prove that if we can't diagonalize against a reduction by forcing nontotality or for it to be incorrect on some fixed input, then we can hyperarithmetically in C_i (or $B \oplus C_i$) recover what the current condition determines the outputs of this reduction to be. Hence, our assumption that D_i is not hyperarithmetic in C_i (or $B \oplus C_i$) means that any reduction we can't diagonalize against won't turn out to compute D_i.

Additionally, we must preserve $\omega_1^{C_i}$ or $\omega_1^{B \oplus C_i}$, depending on the case. For both, we force over nonstandard models $\mathcal{M}_i$ of ZFC, in particular, we force over countable ω-models omitting ω_1^{CK}, yet containing C_i (or $B \oplus C_i$). Harrington, Shore, and Slaman [9] have shown that we can produce such models which are strictly hyperarithmetic in $\mathcal{O}$. As such, $\mathcal{O}$ can determine which $X \in \mathbf{X}_{p_n}$ are not in $\mathcal{M}_i$, and so can produce a modified condition p'_n which has forgotten each real not appearing in the model. Furthermore, $\mathcal{O}$ can enumerate each element of the model which is a dense subset of (the model's version of) Kumabe-Slaman forcing. Then $\mathcal{O}$ can search for an extension of p'_n in $\mathcal{M}_i$ which meets a dense set appearing in the model. By the compatibility lemma, there is such an extension which adds no new computations to any of the reals we have forgotten, and so we can pick such an extension q and extend the master sequence by defining $p_{n+1} = (\Phi_q, \mathbf{X}_q \cup \mathbf{X}_p)$, which does not interfere with the coding procedure.

Hence, for each i, the master sequence induces a sequence $p'_n = (\Phi_{p_n}, \mathbf{X}_{p_n} \cap (2^\omega)^{\mathcal{M}_i})$ such that each $p'_n \in \mathcal{M}_i$ and the sequence $\{p'_n\}$ is $\mathcal{M}_i$-generic for $\mathcal{M}_i$'s Kumabe-Slaman forcing. Thus, on general grounds, the generic object G' corresponding to $\{p_n\}$ preserves ω_1^{CK} and, indeed, as $C_i \in \mathcal{M}_i$ we even have $C_i \oplus G'$ preserves ω_1^{CK}. Note, though, that the generic object G' is the same object as G, the generic for the master sequence, and so, $C_i \oplus G$ preserves ω_1^{CK} as required.

Finally, we need to avoid ideals below E_j. Previously we gave two arguments for this; one via counting, and one directly diagonalizing. We can't use counting so we have to show we can diagonalize effectively in $\mathcal{O}$. Given E_j, $\mathcal{O}$ can compute each set $Y \leq_h \mathcal{O}$ because $\mathcal{O}$ can enumerate and compute all the E_j-hyperarithmetic reductions. Suppose we have a condition $(\Phi_p, \mathbf{X}_p)$, a set Y, and we are diagonalizing against $\Phi_g = Y$. We can assume that Y is a use monotone Turing functional which is correct for B on input A, as Φ_g will be such an object, and $\mathcal{O}$ can determine whether Y is such a set for each $Y < \mathcal{O}$. Suppose that every n we want to put into Φ_p (i.e. $n \notin Y$ but is allowed to enter Φ_p) would interfere with our coding procedure, i.e., is of the form (x, y, α) with $\alpha \subset A$. As $\mathbf{X}_p$ does not contain A (by induction) there is some sufficiently long initial segment of A not an initial segment of any $X \in \mathbf{X}_p$ (and sufficiently long so as to not mess with use monotonicity, or that Φ_p is a Turing functional, and so on). Let x be the least number such that there is no axiom about x applying to A in Φ_p (i.e., is the next value we need to code). As Y is a use monotone Turing functional correct for B on input A there is only one axiom $(x, y, \alpha) \in Y$ with $\alpha \subset A$, so all we need to do is put in (x, y, α') where $\alpha' \subset A$ is sufficiently long to be allowed, and is not precisely α. This information can all be determined uniformly in $\mathcal{O}$ and so we can diagonalize by meeting appropriate dense sets.

Consequently, we can produce a generic of the correct kind hyperarithmetically in $\mathcal{O}$ as required.

3.4 Remarks

Firstly, we observe that the methods of Section 3.2 apply to the arithmetic degrees. For simple end extensions the generic you construct only needs to decide sentences of finite rank, and there is no need to try to preserve any ordinals. For free extensions, Cohen forcing suffices. See Odifreddi [17] for arithmetical Cohen forcing. Consequently, we can always extend USL embeddings into $\mathcal{D}_a$ if the extension is a finite end extension. Furthermore Simpson [22] has shown that every finite lattice is an initial segment of the arithmetic degrees. Consequently, we have now fully solved the finite extension of embeddings problems for USLs into $\mathcal{D}_a$ and so decided its Σ_2 theory as a USL.

Secondly, Jockusch and Slaman [10] answer the extension of embeddings question for countable USLs into $\mathcal{D}_T$. Their positive half uses the full strength of Theorem 3.3 (with countably many parameters) to prove a quantifier elimination theorem in the style of Theorem A.3. The author believes that this argument can be adapted to the hyperarithmetic setting.

However, countable $\mathrm{USL}^\top$ embeddings into $\mathcal{D}_h(\leq_h \mathcal{O})$ are quite different. $\mathcal{D}_h(\leq_h \mathcal{O})$ is itself a $\mathrm{USL}^\top$ and so embeds in itself via the identity. This can't be extended to any superstructure as the embedding is surjective.

CHAPTER 4

INITIAL SEGMENTS OF THE HYPERDEGREES

The previous chapter provided the "positive" half of the extension of embedding problems for finite USLs into $\mathcal{D}_h$ and for finite USL$^\top$s into $\mathcal{D}_h(\leq_h \mathcal{O})$ by saying that we can always extend embeddings as long as the extension is an end extension or an almost end extension, respectively. The "negative" half requires us to show that embeddings do not always extend if the extension is not an (almost) end extension, i.e., given $\mathscr{U}$ and $\mathscr{V}$ extending it, we want an embedding of $\mathscr{U}$ that does not extend to $\mathscr{V}$. Our solution is to provide an embedding of $\mathscr{U}$ as an (almost) initial segment of $\mathcal{D}_h$.

Recall that a finite USL$^\top$ is a bounded lattice. A USL$^\top$ embedding of a lattice into $\mathcal{D}_h(\leq_h \mathcal{O})$ need not preserve the meet structure of the lattice. However, if the range of the embedding is an initial segment (or almost initial segment), then the meet structure must be preserved. Consequently, this chapter concerns bounded lattice embeddings. For the rest of this chapter lattice means bounded lattice.

Kjos-Hanssen and Shore [11] have shown that every sublattice of every hyperarithmetic lattice embeds as an initial segment of $\mathcal{D}_h$. This leaves the question of embeddings into $\mathcal{D}_h(\leq_h \mathcal{O})$. In this section we focus on almost initial segment embeddings of finite lattices into $\mathcal{D}_h(\leq_h \mathcal{O})$. Our proof uses Kjos-Hanssen and Shore's framework to produce initial segments combined with the coding apparatus of Lerman and Shore [16] who produced almost initial segments of $\mathcal{D}_T(\leq_T 0')$.

Theorem 4.1. *If $\mathscr{L}$ is a finite lattice, then there is a lattice embedding of $\mathscr{L}$ into $\mathcal{D}_h(\leq_h \mathcal{O})$ as an almost initial segment, i.e., there is an injective map from $\mathscr{L}$ into $\mathcal{D}_h(\leq_h \mathcal{O})$ that preserves $\sqsubseteq, \sqcup, \sqcap, \bot, \top$, and the image of $\mathscr{L} \setminus \{\top\}$ is an initial*

segment of $\mathcal{D}_h(\leq_h \mathcal{O})$.

Let $\mathscr{L}$ be a finite lattice, and let A be the set of coatoms of $\mathscr{L}$, i.e.,

$$A = \{x \in \mathscr{L} : x \sqsubset \top \text{ and there is no } y \in \mathscr{L} \text{ s.t. } x \sqsubset y \sqsubset 1\}.$$

Let f be an embedding of $\mathscr{L}$ into $\mathcal{D}_h$ as an initial segment, as provided by Kjos-Hanssen and Shore [11]. As our lattice is finite, it is recursive; therefore, we can choose such an f with takes $\top$ to a degree below that of $\mathcal{O}$. We define a map $\tilde{f} : \mathscr{L} \to \mathcal{D}_h(\leq_h \mathcal{O})$ by

$$\tilde{f}(x) = \begin{cases} f(x) & \text{if } x \neq \top, \\ \text{degree}(\mathcal{O}) & \text{if } x = \top. \end{cases}$$

This map will not, in general, be a lattice embedding of $\mathscr{L}$, as we may no longer preserve the join structure. However, if $|A| < 2$, then there are no $x, y \in \mathscr{L}$ such that $x, y < \top$, yet $x \sqcup y = \top$. In this case, $\tilde{f}$ will be a lattice embedding and, by choice of f, will be an almost initial segment embedding of $\mathscr{L}$ into $\mathcal{D}_h(\leq_h \mathcal{O})$. (Indeed, we could send $\top$ to any degree above $f(\top)$ and still have a lattice embedding where the image of $\mathscr{L} \setminus \{\top\}$ is an initial segment). Henceforth, we assume that $|A| \geq 2$.

The rest of this section approximately follows the structure of Kjos-Hanssen and Shore [11] with the additional concern of coding $\mathcal{O}$ into the image of the top element of a lattice. Our notion of forcing is rather simpler than theirs (because we are only concerned with finite lattices), but whenever we wish to meet a dense set of a particular kind we need to show we can do so without interfering with our coding procedure (which we are yet to define).

4.1 Lattice representations

To define our notion of forcing for almost initial segments we require a strong kind of representation of $\mathscr{L}$.

Definition 4.2 (USL table)**.** A set Θ of maps from $\mathscr{L}$ to ω is an **USL table** for $\mathscr{L}$ if it has the following properties:

(1) (Nontriviality of Θ) The zero map $x \mapsto 0$ is in Θ (we denote this map by 0 as well).

(2) ($\perp$ is trivial) For every $\alpha \in \Theta$, $\alpha(\perp) = 0$.

And for every choice $x, y, z \in \mathscr{L}$:

(3) (Order) If $x \sqsubseteq y$, and $\alpha, \beta \in \Theta$ satisfy $\alpha(y) = \beta(y)$, then $\alpha(x) = \beta(x)$.

(4) (Differentiation) If $x \not\sqsubseteq y$, then there are $\alpha, \beta \in \Theta$ such that $\alpha(y) = \beta(y)$, yet $\alpha(x) \neq \beta(x)$.

(5) (Join) If $x \sqcup y = z$, and if $\alpha, \beta \in \Theta$ satisfy $\alpha(x) = \beta(x)$ and $\alpha(y) = \beta(y)$, then $\alpha(z) = \beta(z)$.

Notation. We will denote lattice tables by $\Theta, \Theta_1, \Theta_2$ and so on, and their elements will be denoted by lowercase Greek letters α, β and γ.

For $x \in \mathscr{L}$ and $\alpha, \beta \in \Theta$ members of an USL table for $\mathscr{L}$, we write $\alpha \equiv_x \beta$ if $\alpha(x) = \beta(x)$, which is clearly an equivalence relation on Θ. We write $\alpha \equiv_{x,y} \beta$ to mean that $\alpha \equiv_x \beta$ and $\alpha \equiv_y \beta$.

We extend this notation to partial functions (and so, in particular, to strings) by declaring $f \equiv_x g$ if wherever f, g are both defined their values agree modulo x.

Definition 4.3 (Sequential lattice representation)**.** A nested sequence $\{\Theta_i : i \in \omega\}$ of finite USL tables for $\mathscr{L}$ is a **sequential (lattice) representation** for $\mathscr{L}$ if for every $i \in \omega$:

(1) There are **meet interpolants** for Θ_i in Θ_{i+1}, i.e., if $x, y, z \in \mathscr{L}$ satisfy $x \sqcap y = z$, and if $\alpha, \beta \in \Theta_i$ satisfy $\alpha \equiv_z \beta$, then there are $\gamma_0, \gamma_1, \gamma_2 \in \Theta_{i+1}$ such that

$$\alpha \equiv_x \gamma_0 \equiv_y \gamma_1 \equiv_x \gamma_2 \equiv_y \beta.$$

(2) There are **homogeneity interpolants** for Θ_i in Θ_{i+1}, i.e., for all α_0, α_1, β_0,and $\beta_1 \in \Theta_i$ such that

$$(\forall x \in \mathscr{L})\,[\alpha_0 \equiv_x \alpha_1 \to \beta_0 \equiv_x \beta_1],$$

and there are $\gamma_0, \gamma_1 \in \Theta_{i+1}$ and $\mathscr{L}$**-homomorphisms** $f, g, h : \Theta_i \to \Theta_{i+1}$ such that

$$f : \alpha_0, \alpha_1 \mapsto \beta_0, \gamma_1, \quad g : \alpha_0, \alpha_1 \mapsto \gamma_0, \gamma_1, \quad h : \alpha_0, \alpha_1 \mapsto \gamma_0, \beta_1.$$

(f is an $\mathscr{L}$-homomorphism from Θ_i to Θ_{i+1} if for each $\alpha, \beta \in \Theta_i$ and each $x \in \mathscr{L}$ if $\alpha \equiv_x \beta$, then $f(\alpha) \equiv_x f(\beta)$.)

The above definition is a simplification of the representation given in Theorem 5.1 of Kjos-Hanssen and Shore [11]. Ours is simpler because our lattices are finite, and so we do not need to approximate our lattice with a growing sequence of finite substructures. Using such a representation we could embed $\mathscr{L}$ as an initial segment of $\mathcal{D}_h$, or even $\mathcal{D}_h(\leq_h \mathcal{O})$; however, we would have insufficient control over the image of $\top$. We introduce apparatus that allows us to code $\mathcal{O}$ into the image of $\top$:

Definition 4.4 (Coding-ready representation)**.** A sequential representation $\{\Theta_i : i \in \omega\}$ has $C \subseteq \Theta_0$ as a **coding set** if there is a bijective map g from

$$\{\langle x, y, k\rangle : k \in \{0, 1\},\ x \sqcup y = \top, \text{ and } x, y \neq \top\}$$

to C such that:

(3) For every $x, y \in \mathscr{L} \setminus \{\top\}$ if $x \sqcup y = \top$, then

$$g(x, y, 0) \equiv_x g(x, y, 1) \text{ and } g(x, y, 0) \not\equiv_y g(x, y, 1).$$

(4) For all $\alpha \in C$ and all $x \in \mathscr{L} \setminus \{\top\}$, there is a $\beta \in \Theta_0 \setminus C$ such that $\alpha \equiv_x \beta$.

The table is **acceptable for** A (the set of coatoms of $\mathscr{L}$) if for each $i > 0$ there is a subset Θ_i^* of Θ_i, containing Θ_{i-1} such that (1) holds for Θ_{i+1}^* in place of Θ_{i+1}, (2) holds for Θ_{i+1}^* in place of Θ_i, and further, (in the notation of (2)) if $f(\alpha) \in C$, then $\alpha = \alpha_0$ or $\alpha = \alpha_1$ (and the same for g and h), and, finally:

(5) For all $x \in A$, all $i \in \omega$, and all $\alpha_0, \alpha_1 \in \Theta_i$ there exists $\beta_0, \beta_1 \in \Theta_{i+1}^* \setminus \Theta_i$ such that

$$\alpha_0 \equiv_x \beta_0,\ \alpha_1 \equiv_x \beta_1, \text{ and } (\forall y \in \mathscr{L})[\alpha_0 \equiv_y \alpha_1 \to \beta_0 \equiv_y \beta_1].$$

If $\{\Theta_i : i \in \omega\}$ has coding set C, the differentiation property of USL tables is satisfied outside of C (i.e., for each $x \not\sqsubseteq y$ there are $\alpha, \beta \in \Theta_0 \setminus C$ such that $\alpha \equiv_y \beta$ yet $\alpha \not\equiv_x \beta$), and it is acceptable for A, we call it **coding ready**.

We will construct a recursive coding ready sequential representation $\{\Theta_i : i \in \omega\}$ for $\mathscr{L}$ shortly. Firstly, we motivate and explain the definition: our representation will be nested as displayed:

$$C \subsetneq \Theta_0 \subsetneq \Theta_1^* \subsetneq \Theta_1 \subseteq \Theta_2^* \subsetneq \Theta_2 \subsetneq \cdots$$

Each Θ_i will be a USL table for $\mathscr{L}$ such that we can find meet interpolants for elements of Θ_i inside Θ^*_{i+1}, and we can find homogeneity interpolants for Θ^*_{i+1} in Θ_{i+1}. The elements of C are special, and they indicate we are coding. Clause (3) tells us that for each $x, y \in \mathscr{L}$ joining up nontrivially to $\top$ that there is a pair of distinguished coding elements. Clause (4) and the fact that we can satisfy the differentiation property outside of C tells us that we can replace a coding element with an element that does not code and still preserve a congruence. Clause (5) in the definition of acceptable for A tells us that if we can find a split (this will be defined later), then we can find a split not using coding elements, and the requirement that f, g, and h only take on coding values when entirely necessary (i.e. when $\beta_0 \in C$ or $\beta_1 \in C$) allows us to find homogeneity interpolants which do no coding.

Theorem 4.5. *Let $\mathscr{L}$ be a finite lattice with at least two coatoms, and let A be the set of coatoms of $\mathscr{L}$. Then there is a recursive coding ready sequential lattice representation for $\mathscr{L}$.*

Lerman and Shore [16] construct a sequential representation which is very similar to ours; the main difference is in the homogeneity interpolants. In our notation the $\mathscr{L}$-homomorphisms f, g, h in Lerman and Shore act on α_0, α_1 as follows:

$$f : \alpha_0, \alpha_1 \mapsto \beta_0, \gamma_0, \quad g : \alpha_0, \alpha_1 \mapsto \gamma_0, \gamma_1, \quad h : \alpha_0, \alpha_1 \mapsto \gamma_1, \beta_1.$$

In particular, $f(\alpha_1) = \gamma_0$ and $h(\alpha_0) = \gamma_1$ instead of γ_1 and γ_0, respectively, which is what we required in (2) of the definition of sequential representation. (There is also a difference in the coding set; Lerman and Shore have coding elements for each unordered pair $\{x, y\}$ and we have them for each ordered pair $\langle x, y\rangle$. The reason for this difference is purely notational, and does not present any mathematical difficulties.)

In their construction Lerman and Shore begin with a finite USL table Θ for $\mathscr{L}$ and then construct a finite USL table extension Θ_0 of Θ and observe that Θ_0 contains a coding set C disjoint from Θ satisfying properties (3) and (4). Furthermore, as we started with a USL table Θ, the differentiation property is satisfied outside C, as required.

Then they proceed inductively: Given Θ_i, by Lerman Appendix B.2.6 [15], there is a finite USL table Θ_i^1 which contains meet interpolants for Θ_i. They then argue that Θ_i^1 can be extended to a finite USL table Θ_i^* satisfying (5), and as $\Theta_i \subseteq \Theta_i^1 \subsetneq \Theta_{i+1}^*$ we can find meet interpolants for Θ_i in Θ_{i+1}^*. All of this is uniformly recursive.

Consequently, all we need to show is that given Θ_{i+1}^* a finite USL table for $\mathscr{L}$ that we can find (uniformly and recursively) a finite USL table extension Θ_{i+1} of Θ_{i+1}^* such that we can find homogeneity interpolants for Θ_{i+1}^* in Θ_i and the $\mathscr{L}$-homomorphisms avoid the coding set (unless $\beta_0 \in C$, or $\beta_1 \in C$).

Kjos-Hanssen and Shore [11] have already completed this work for us: Their Proposition 5.6 (taking $\hat{\mathscr{L}} = \mathscr{L}$) says that we can find such a USL table extension which has homogeneity interpolants of the kind we need (again uniformly and recursively). An examination of their proof shows that if $\alpha \in \Theta_{i+1}^*$, then $f(\alpha) \notin \Theta_{i+1}^*$ unless $\alpha = \alpha_0$, and so $f(\alpha) = f(\alpha_0) = \beta_0$, and similarly for g and h. Hence applying this Proposition allows us to continue our induction and completes the construction.

4.2 Perfect trees and forcing

From here onward we fix a finite lattice $\mathscr{L}$ with a set of coatoms A of cardinality at least two and fix a recursive coding ready sequential representation $\{\Theta_i : i \in \omega\}$ for $\mathscr{L}$.

Definition 4.6 (Uniform tree)**.** A **uniform tree** for the representation $\{\Theta_i : i \in \omega\}$ is a function T with both domain and range the set of all strings σ such that if $\sigma(n)$ is defined, then $\sigma(n) \in \Theta_n$, which satisfies the following properties for all $\sigma, \tau \in \operatorname{dom} T$:

(1) (Order) If $\sigma \subseteq \tau$, then $T(\sigma) \subseteq T(\tau)$.

(2) (Nonorder) If $\sigma | \tau$, then $T(\sigma) | T(\tau)$. In fact, we require that for each length l there is a string π such that if $|\sigma| = l$ and $\alpha \in \Theta_l$ (so that $\sigma^\frown\alpha \in \operatorname{dom} T$), then $T(\sigma^\frown\alpha) \supseteq T(\sigma)^\frown\pi^\frown\alpha$.

(3) (Uniformity) For every fixed length l, there is a string π_l and for every $\alpha \in \Theta_l$, there is a string $\rho_{l,\alpha}$ whose length does not depend on α such that if $|\sigma| = l$, then $T(\sigma^\frown\alpha) = T(\alpha)^\frown\pi_l{}^\frown\rho_{l,\alpha}$. Note, we require that π_l and $\rho_{l,\alpha}$ only depend on the length of σ.

We say T is **branch coding free** if for every length l and every $\alpha \in \Theta_l$, the string $\pi_l{}^\frown\rho_{l,\alpha}$, which is the extension corresponding to α at this level, does not do **unnecessary coding**, i.e., for each j if $(\pi_l{}^\frown\rho_{l,\alpha})(j) \in C$, then $j = |\pi_l|$ (and so $\alpha \in C$). (This means that the only time a member α of our coding set C appears on a branch is when we are at a fork and we took the path corresponding to α).

The tree T is **congruence respecting** if for every length l, every $\alpha, \beta \in \Theta_l$, and every $x \in \mathscr{L}$ if $\alpha \equiv_x \beta$, then $\pi_l{}^\frown\rho_{l,\alpha} \equiv_x \pi_l{}^\frown\rho_{l,\beta}$ (or, equivalently, $\rho_{l,\alpha} \equiv_x \rho_{l,\beta}$).

Our trees are related to those in Definition 2.4 of Kjos-Hanssen and Shore [11]. We are afforded some simplifications, again, because our lattice is finite; we can preserve all congruences all the time, and have the same domain for all our trees. Our nonorder (and, consequently, uniformity) property is different in that we have the string π that every branch at a level has to follow before splitting. This is a technical requirement that allows us to prove the fusion lemma (Kjos-Hanssen and Shore also need this modification, their fusion lemma, as written, does not produce a forcing condition). The requirement that the trees are branch coding free is new and allows us to code.

Notation. Uniform trees will be denoted by uppercase Roman letters, most frequently T, S, R, and we denote the set of branches of T by $[T]$. Strings of members of the lattice representation will be denoted σ, τ, ρ, ν and so on, we reserve π for the π in the uniformity property. We write the concatenation of σ by τ as $\sigma^\frown\tau$ and we confuse a string of length one with its value. So, for instance, we may write $\sigma^\frown\alpha$ for $\sigma \in \prod_{i=0}^{l} \Theta_i$ and $\alpha \in \Theta_{n+1}$.

For technical reasons, we define the **height** of a level l to be $|T(\sigma)^\frown\pi_l|$ where σ is any string of length l, and π_l is the string as in the definition of the uniformity property. By uniformity, this is independent of the choice of σ and so is well-defined.

Hyperarithmetic, branch coding free, congruence respecting, uniform trees will be the conditions of our notion of forcing for producing almost initial segments. Observe that the identity tree satisfies all these properties.

Definition 4.7 (Subtree)**.** A uniform tree S is a **subtree** of a uniform tree T, written $S \subseteq T$, if the range of S is contained in the range of T.

We single out two operations on uniform trees.

Definition 4.8. If T is a uniform tree and $\sigma \in \prod_{i=0}^{l} \Theta_i$ for some l, then we define T_σ by:

$$T_\sigma(\tau) = T(\sigma^\frown \tau).$$

If $\mu \in \prod_{i=0}^{l} \Theta_i$ for some l and $l \leq |T(\emptyset)| - 1$, then the **transfer tree of T over μ**, written T^μ, is the tree such that $T^\mu(\sigma)$ is the string $T(\sigma)$ but with its initial segment of length l replaced by μ. (i.e., you change the root of T by replacing the initial segment of the right length by μ). We write T^μ_σ for $(T_\sigma)^\mu$.

Proposition 4.9. *Let T be a uniform tree. Then T_σ and T^μ are uniform trees whenever they are defined. Furthermore, if T is branch coding free, congruence respecting, or hyperarithmetic, then T_σ and T^μ are, correspondingly, branch coding free, congruence respecting, or hyperarithmetic. Finally, T_σ is a subtree of T whenever it is defined, and $(T_\sigma)_\tau = T_{\sigma^\frown\tau}$.*

Proofs of these claims are entirely routine and are omitted.

Definition 4.10 (Perfect forcing)**.** **$\{\Theta_i : i \in \omega\}$-perfect forcing** is the set $\mathcal{P}_{\{\Theta_i : i \in \omega\}} = \mathcal{P}$ of all hyperarithmetic, branch coding free, congruence respecting, uniform trees ordered by the subtree relation, i.e., for $T, S \in \mathcal{P}$ we say T extends S or T refines S, written $T \leq_\mathcal{P} S$, if T is a subtree of S.

Now we have our notion of forcing we can discuss the objects its conditions approximate. Clearly, for each length l the set $\{T \in \mathcal{P} : |T(\emptyset)| > l\}$ is dense in $\mathcal{P}$. Consequently, a descending sequence of conditions $\{T_i\}_{i=0}^{\infty}$ meeting these dense sets will correspond to an object $\mathcal{G}$: an element of the product $\prod_{i=0}^{\infty} \Theta_i$, defined by $\mathcal{G}(n) = \alpha$ iff there is an i such that $T_i(\emptyset)(n) \downarrow = \alpha$. For each $x \in \mathscr{L}$ we define $\mathcal{G}^x$,

an element of ω^ω, by

$$\mathcal{G}^x(n) = \mathcal{G}(n)(x).$$

Our embedding will take $x \in \mathscr{L}$ to the degree of $\mathcal{G}^x$. To ensure that $\mathcal{G}^\top \geq_h \mathcal{O}$ we need to do some coding.

Definition 4.11 (Coding)**.** Fix $x, y \in \mathscr{L}$ which join up nontrivially to $\top$ (i.e., $x \sqcup y = \top$ and $x, y \neq \top$). The **root coding of T for $\langle x, y\rangle$** is the number of occurrences of $g(x, y, 0)$ and $g(x, y, 1)$ in the string $T(\emptyset)$. A subtree S of T does **no more root coding than T for $\langle x, y\rangle$** if the root coding of T for $\langle x, y\rangle$ is the same as that for S and it does **no more root coding** if it does no more root coding for each pair $\langle x, y\rangle$ which join up nontrivially to $\top$.

Given $\mathcal{G}^x \oplus \mathcal{G}^y$ our decoding procedure is as follows:

> On input n search for the nth number m such that $\mathcal{G}(m) \equiv_x g(x, y, 0)$ and either $\mathcal{G}(m) \equiv_y g(x, y, 0)$ or $\mathcal{G}(m) \equiv_y g(x, y, 1)$. If $\mathcal{G}(m) \equiv_y g(x, y, 0)$ we say n is not in the set, and if $\mathcal{G}(m) \equiv_y g(x, y, 1)$ we say n is in the set.

(Note that g is the function in the definition of coding set, and that the decoding procedure for $\mathcal{G}^x \oplus \mathcal{G}^y$ depends on the order of x and y.)

This procedure is clearly recursive in $\mathcal{G}^x \oplus \mathcal{G}^y$, as $\{\Theta_i : i \in \omega\}$ is recursive, and to determine whether $\mathcal{G}(m) \equiv_z \alpha$ it suffices to know $\mathcal{G}^z(m)$. Also, our decoding procedure does not detect noncoding elements: If $\gamma \equiv_x g(x, y, 0)$, then $\gamma \equiv_x g(x, y, 1)$ too. Hence, if, further, $\gamma \equiv_y g(x, y, 0)$, then, by the join property, $\gamma \equiv_{x \sqcup y} g(x, y, 0)$. But, as $x \sqcup y = \top$, $\gamma \equiv_\top g(x, y, 0)$ and so $\gamma = g(x, y, 0)$. Entirely similarly, if $\gamma \equiv_y g(x, y, 1)$, then $\gamma = g(x, y, 1)$.

Hence, to ensure that our decoding procedure gives the characteristic function of some set X, it suffices to construct a $\mathcal{G}$ such that, for each n, at the nth place where $\mathcal{G}(m) = g(x, y, 0)$ or $\mathcal{G}(m) = g(x, y, 1)$, that it is $g(x, y, 0)$ iff $n \notin X$ and is $g(x, y, 1)$ iff $n \in X$. Therefore, a running theme for the remainder of this section will be meeting dense sets of various kinds without increasing the root coding of a condition.

4.3 The forcing relation

With this strategy in mind we must define our language of forcing and our forcing relation, and show we can provide a sufficient degree of genericity without interfering with the coding procedure. Our language and model are the same as for Cohen forcing outlined in Section 2.5

Definition 4.12 (Forcing relation)**.** Let φ be a sentence of $\mathcal{L}(\omega_1^x, \mathtt{G})$ and T a forcing condition. We define $T \Vdash_x \varphi$ by induction:

(1) If φ is ranked, then $T \Vdash_x \varphi$ iff for every $\mathcal{G} \in [T]$, $\mathcal{M}(\omega_1^x, G)$ satisfies φ.

(2) If φ is unranked and $\varphi = (\exists n)\psi(n)$, then $T \Vdash \varphi$ iff there is an $n \in \omega$ such that $T \Vdash \psi(\underline{n})$.

(3) If φ is unranked and $\varphi = (\exists X^\delta)\psi(X^\delta)$, then $T \Vdash_x \varphi$ iff there is a term $\mathcal{H}(n)$ of rank at most δ such that $T \Vdash_x \varphi(\hat{n}\mathcal{H}(n))$.

(4) If φ is unranked and $\varphi = (\exists X)\psi(X)$, then $T \Vdash_x \varphi$ iff there is a $\delta < \omega_1^{\mathrm{CK}}$ such that $T \Vdash_x (\exists X^\delta)\psi(X^\delta)$.

(5) If φ is unranked and $\varphi = \psi_1 \wedge \psi_2$, then $T \Vdash_x \varphi$ iff $T \Vdash_x \psi_1$ and $T \Vdash_x \psi_2$.

(6) If φ is unranked and $\varphi = \neg\psi$, then $T \Vdash_x \varphi$ iff for all $S \in \mathcal{P}$ extending T, $\neg S \Vdash_x \psi$.

Notation. We denote formulas of our forcing languages as φ, ψ, and occasionally use $\mathcal{H}(n)$ for a ranked formula with only one free variable, n, which is a natural number variable.

As $\Vdash_x$ only holds between forcing conditions and sentences in $\mathcal{L}(\omega_1^x, \mathtt{G})$, we shall omit the x from the $\Vdash$ if we have declared from which language the sentences come.

It is standard to define forcing to be equal to truth for atomic formulas of a forcing language. However, we treat all ranked formulas at this ground level and define forcing to be equal to truth for all of them. This obscures the fact, which we will need to establish, that given a condition T and a sentence φ there is an extension of T deciding φ.

Definition 4.13 (Generic sequence)**.** A sequence $\{T_i : i \in \omega\}$ of elements of $\mathcal{P}$ is $\mathcal{L}(\omega_1^x, \mathtt{G})$-**generic** if $T_{i+1} \leq_{\mathcal{P}} T_i$ for each i, and for every $\varphi \in \mathcal{L}(\omega_1^x, \mathtt{G})$ there is an i such that T_i forces either φ or $\neg\varphi$. The sequence is **generic** if it is generic for each $x \in \mathscr{L} \setminus \{\top\}$.

Observe that if $\{T_i : i \in \omega\}$ is generic, then it meets each dense set

$$D_n = \{T \in \mathcal{P} : T(\emptyset)(n)\downarrow\},$$

and so there is a unique $\mathcal{G} \in \bigcap_{i\in\omega}[T_i]$. We call such a $\mathcal{G}$ the **generic**.

Lemma 4.14 (Standard lemmas)**.** *Let T be a condition, $x \in \mathscr{L} \setminus \{\top\}$, and $\varphi \in \mathcal{L}(\omega_1^x, \mathtt{G})$. Then the following hold:*

(1) (Consistency) $T \nVdash \varphi \wedge \neg\varphi$.

(2) (Extension) If S extends T and $T \Vdash \varphi$, then $S \Vdash \varphi$.

(3) (Density) There is an S extending T deciding φ (i.e., S either forces φ or forces its negation).

(4) (Forcing and truth) If $\{T_i : i \in \omega\}$ is x-generic and $\mathcal{G}$ is the generic object, then $\mathcal{M}(\omega_1^x, G) \vDash \varphi$ iff there is an i such that $T_i \Vdash \varphi$.

Proof of the consistency and extension properties. First the consistency property: Suppose φ is ranked. Then it follows from the fact that $\mathcal{M}(\omega_1^x, G) \nvDash \varphi \wedge \neg\varphi$ that T does not force both φ and its negation. If φ is unranked, then the definition of forcing for negation implies that T does not force both φ and its negation.

For the extension property; if φ is ranked and $S \leq_{\mathcal{P}} T$, then $[S] \subseteq [T]$, consequently, if every branch of T satisfies φ, then every branch of S does too, and so $S \Vdash \varphi$. If φ is unranked, we proceed by induction on the full ordinal rank of φ. Details can be found in Chapter IV Section 4 of Sacks [19]. □

The inductive step in a standard proof of the density property goes through as normal (using the definition of forcing for negation). The issue is with the base case: It is not clear that given a condition that there is a refinement such that every branch satisfies a particular ranked formula. To show the existence of such a condition we need to establish the, so called, fusion property of trees. We also have the concern of coding unnecessarily at the root.

Before establishing the fusion lemma, we need some technical facts about the forcing relation.

Lemma 4.15. *The relation $T \Vdash \varphi$ restricted to Σ^1_1 sentences $\varphi \in \mathcal{L}(\omega_1^x, \mathcal{G})$ is Π^1_1.*

Proof. Our situation is only slightly different from that in Sacks Chapter IV, Lemma 4.2. We have a slightly different notion of tree and of extension, but they are all uniformly arithmetic in codes for the trees and so the complexity has not increased. □

Definition 4.16. Let $\sigma \in \prod_{i=0}^{l} \Theta_i$ be a string and $x \in \mathscr{L} \setminus \{\top\}$. The **$x$-safe version of** σ, written σ_x, is defined by taking each $n < |\sigma|$ such that $\sigma(n) = \alpha \in C$ and defining $\sigma_x(n) = \beta$ where $\beta \in \Theta_0 \setminus C$ is the member of the lattice table which agrees with α modulo x, but is not coding, and otherwise not changing σ. (There may be more than once such β, so for each $x \in \mathscr{L} \setminus \{\top\}$ and coding α pick a particular β and always use that.) Observe that $\sigma \equiv_x \sigma_x$.

If T is a condition and S extends T, then the **x-safe version of S with respect to** T is the condition $S^{T(\sigma_x)}$, where σ is the string such that $T(\sigma) = S(\emptyset)$, and σ_x is its x-safe version. Note that $\{\mathcal{G}^x : \mathcal{G} \in [S]\} = \{\mathcal{G}^x : \mathcal{G} \in [S^{T(\sigma_x)}]\}$ (as T preserves congruences, and so $T(\sigma) \equiv_x T(\sigma_x)$). Also note the x-safe version of S with respect to T does no more root coding than T (as $T(\sigma_x) = S^{T(\sigma_x)}(\emptyset)$, σ_x does no coding, and T is branch coding free).

Lemma 4.17. *Let S and S' be conditions that have the same branches modulo x (i.e., $\{\mathcal{G}^x : \mathcal{G} \in S\} = \{\mathcal{G}^x : \mathcal{G} \in S'\}$), and let φ be a sentence of $\mathcal{L}(\omega_1^x, \mathcal{G})$. Then $S \Vdash \varphi$ iff $S' \Vdash \varphi$. Hence, in particular, if there is an extension S of T forcing a sentence, then there is an extension of T forcing that sentence which does no more root coding than T: namely, the x-safe version of S.*

Proof. Base case, φ is ranked: By the definition of forcing for ranked formulas, as the branches of S and the branches of S' are the same modulo x, then every branch of S satisfies φ iff every branch of S' does.

Inductive step: If $\varphi = \psi_1 \wedge \psi_2$, then $S \Vdash \varphi$ iff $S \Vdash \psi_1$ and $S \Vdash \psi_2$ iff (by induction) $S' \Vdash \psi_1$ and $S' \Vdash \psi_2$ iff $S' \Vdash \psi_i \wedge \psi_2$.

If $\varphi = (\exists x)\psi(x)$, then $S \Vdash \varphi$ iff there is an n such that $S \Vdash \psi(\underline{n})$ iff there is an n such that $S' \Vdash \psi(\underline{n})$ iff $S' \Vdash (\exists x)\psi(x)$.

For $\varphi = (\exists X^\delta)\psi(X^\delta)$, the proof is similar to the natural number existential, but with a witnessing formula $\mathcal{H}$ of rank at most δ in place of n.

If $\varphi = (\exists X)\psi(X)$, then $S \Vdash \varphi$ iff there is a $\delta < \omega_1^{\mathrm{CK}}$ such that $S \Vdash (\exists X^\delta)\psi(X^\delta)$ iff there is a $\delta < \omega_1^{\mathrm{CK}}$ such that $S' \Vdash (\exists X^\delta)\psi(X^\delta)$ iff $S' \Vdash \varphi$.

The case when $\varphi = \neg\psi$ is somewhat trickier, we need to be able to transform extensions R of S which force ψ into extensions R' of S' also forcing ψ. By the inductive hypothesis, it suffices to ensure that the branches of R' are the same as those of R modulo x.

Given $R \leq_{\mathcal{P}} S$ we define $R'(\sigma) = S'(\tau_\sigma)$ where τ_σ is the (unique) string such that $R(\sigma) = S(\tau_\sigma)$. R' is hyperarithmetic as R, S, and S' are (so we can use R and S to find τ_σ for any σ, and then plug this into S'), and, furthermore, R' is a branch coding free, congruence respecting, uniform subtree of S', as R is for S. It remains to show that the branches of R and R' are the same modulo x.

Claim: For each level l the height of S and of S' are the same.

The height of the 0th levels of S and S' are, respectively $|S(\emptyset)^\frown \pi_0^S|$ and $|S'(\emptyset)^\frown \pi_0^{S'}|$. Every branch on S extends $S(\emptyset)^\frown \pi_0^S$, and so every branch on S' must extend this string modulo x. By the implementation of the nonorder property, the different branches of S at this level all disagree at the $|S(\emptyset)^\frown \pi_0^S|$th place. If $x = \perp$, then there is only one branch on S, S', R, R': The constant 0 branch,

and so the claim is shown. Otherwise $x \neq \perp$, and there are α, β which disagree modulo x, and so there are branches of S which disagree at the $|S(\emptyset)^\frown \pi_0^S|$ place, modulo x.

This must be reflected in S', and so there is splitting at the $|S(\emptyset)^\frown \pi_0^{S'}|$th place on S' therefore, the height of S' at level 0 must be less than or equal to that of S. Interchanging S and S' in this argument shows the heights at the root must be equal.

Now we proceed inductively: For σ of length l we assume $|S(\sigma)^\frown \pi_l^S| = |S'(\sigma)^\frown \pi_l^{S'}|$, hence, it suffices to show that $|\rho_{l,\alpha}^{S\frown} \pi_{l+1}^S| = |\rho_{l,\alpha}^{S'\frown} \pi_{l+1}^{S'}|$. But the argument is similar to the base case: We know there are mod x disagreements in branches of S at $|S(\sigma)^\frown \pi_l^S{}^\frown \rho_{\alpha,l}^S{}^\frown \pi_l^S|$, consequently, there must be mod x disagreements in branches of S' at this place too. Hence, the height of the $l+1$ level of S' is at most the height of the $l+1$st level of S. Interchanging S and S' shows they must be equal.

Claim: For every σ and every level l, $S(\sigma)^\frown \pi_l^S \equiv_x S'(\sigma)^\frown \pi_l^{S'}$.

Every branch of S extends $S(\emptyset)^\frown \pi_0^S$ and so they all agree modulo x. This is reflected in the branches of S' and so $S'(\emptyset)^\frown \pi_0^{S'}$ must agree with the initial segment modulo x.

Then, inductively, if it is true up to level l, then consider $S(\sigma)^\frown \pi_l^S{}^\frown \rho_{l,\alpha}^S{}^\frown \pi_{l+1}^S$ for any $\alpha \in \Theta_l$. By induction, $S(\sigma)^\frown \pi_l^S$ and $S'(\sigma)^\frown \pi_l^{S'}$ agree modulo x and by the last claim they are of the same height. Then, by the implementation of nonorder, the first place where this string is undefined but $S(\sigma)^\frown \pi_l^S{}^\frown \rho_{l,\alpha}^S{}^\frown \pi_{l+1}^S$ is takes value α. As S is congruence respecting, then if $\alpha \equiv_x \beta$, then $\rho_{l,\beta}^S \equiv_x \rho_{l,\alpha}^S$, and so every branch of S (and so of S') which looks like α (modulo x) at this place looks the

same for the rest of the string $\rho^S_{l,\alpha}$ modulo x. Consequently, $\rho^{S'}_{l,\alpha}$ must agree modulo x with $\rho^S_{l,\alpha}$, because S' is also congruence respecting.

Claim: The R and R' above have the same branches modulo x.

If $G \in [R]$ is a path, there is some sequence $\{\sigma_i : i \in \omega\}$ of compatible strings such that $R(\sigma_i)$ converges to G, and $|\sigma_i| = i$. As R is a subtree of S, there is a sequence of compatible strings $\{\tau_i : i \in \omega\}$ such that $S(\tau_i) = R(\sigma_i)$. By the previous claim, $S'(\tau_i) \equiv_x S(\tau_i)$, and so

$$R'(\sigma_i) = S'(\tau_i) \equiv_x S(\tau_i) = R(\sigma_i)$$

and so there is some $G' \in [R']$ such that $G \equiv_x G'$. Interchanging the role of R and R' shows that every branch of R' has a corresponding branch in R which agrees modulo x, hence the branches are the same modulo x as required.

So, suppose $S' \Vdash \neg\psi$, then there is no $R' \leq S'$ such that $R' \Vdash \psi$. If there were $R \leq S$ forcing ψ, then we can construct $R' \leq S'$ which forces ψ (as we can construct R' with the same branches as R modulo x and so, by induction, forcing ψ). Consequently, S must force $\neg\psi$, which completes the proof. □

Lemma 4.18 (Fusion)**.** *Let $\{\varphi_i : i \in \omega\}$ be a hyperarithmetic sequence of Σ^1_1 formulas of $\mathcal{L}(\omega_1^x, \mathcal{G})$, and let T be a condition such that for every $S \leq_{\mathcal{P}} T$ and every $j \in \omega$. Then there is an $R \leq_{\mathcal{P}} S$ such that $R \Vdash \varphi_j$. Then there is a condition $V \leq_{\mathcal{P}} T$ forcing each φ_i which does no more root coding than T.*

Proof. Fix such a sequence $\{\varphi_i : i \in \omega\}$ and a condition T. By the previous lemma, for every $S \leq_{\mathcal{P}} T$ and $i \in \omega$, as there is an $R \leq_{\mathcal{P}} S$ forcing φ, there is one forcing φ and doing no more root coding than S. Consider the predicate

$$R \in \mathcal{P}, \text{ refines } S \in \mathcal{P}, \text{ does no more root coding than } S, \text{ and forces } \varphi_i.$$

As $\mathcal{P}$ and the forcing relation are Π^1_1, and the other clauses are arithmetic, this predicate is uniformly Π^1_1 in R, S, i. By Krisel's uniformization theorem, there is a partial Π^1_1 function which produces R in terms of S and i. We denote this function $R(S, i)$. By assumption, R is total on the conditions S extending T.

We want to construct a single condition V that extends T and forces each φ_i simultaneously. We construct V level by level as well as auxiliary conditions U^l_j where $l \in \omega$ is a level, and j varies between 0 and the number $m(l)$, which is one less than the number of branches of T at level l (i.e., $m(l) = \prod_{k=0}^{l} |\Theta_k| - 1$). Fix a simultaneous hyperarithmetic enumeration α^l_k of each Θ_l, and order strings lexicographically.

Stage 0**:** $V(\emptyset) = T(\emptyset)$.

We define U^0_0 to be $R(T_{\alpha^0_0}, 0)$, a subtree of $T_{\alpha^0_0}$ that forces φ_0 and does no more root coding than $T_{\alpha^0_0}$. By the uniformity of T, $(U^0_0)^{T(\alpha^0_1)}$ is a subtree of $T_{\alpha^0_1}$ and so of T. We define $U^0_1 = R((U^0_0)^{T(\alpha^0_1)}, 0)$, we continue in this fashion across the level defining $U^0_{k+1} = R((U^0_k)^{T(\alpha^0_k)}, 0)$. At the end we have $U^0_{m(0)}$. By uniformity, $(U^0_{m(0)})^{T(\alpha^0_k)}$ is a subtree of U^0_k for every k and, as such, must force φ_0.

We define $V(\alpha) = (U^0_{m(0)})^{T(\alpha)}(\emptyset)$ for each $\alpha \in \Theta_0$. By the uniformity of both T and $U^0_{m(0)}$, V, as defined so far, is uniform and congruence respecting.

To see it is branch coding free, observe that if $V(\alpha)(n) = \beta \in C$ and $n \geq |V(\emptyset)|$, then, by definition, $(U^0_{m(0)})^{T(\alpha)}(\emptyset)(n) = \beta$. If $n < |T(\alpha)|$, then $(U^0_{m(0)})^{T(\alpha)}(\emptyset)(n) = T(\alpha)(n)$, and as T is branch coding free, this means that $\alpha = \beta$ and n is precisely $|T(\emptyset)^\frown \pi^T_0|$, which is not unnecessary coding. Otherwise, $n \geq |T(\alpha)|$. Let k be the first stage such that $U^0_k(\emptyset)(n) \downarrow$. U^0_0 does no more root coding than $T(\alpha^0_0)$, by the choice of R, and so, inductively across the level, U^0_{j+1} does no more root coding

than $(U^0_j)^{T(\alpha^0_{j+1})}$ which does no more root coding than $T_{\alpha^0_{j+1}}$. Therefore, $\beta = \alpha^0_k$ and n must be precisely $|T(\alpha^0_k)^\frown \pi^T_1|$, which is not unnecessary coding.

Stage $l > 0$: Assume we have defined V up to level l, and so far it is branch coding free, congruence respecting, and uniform and that we have $U^{l-1}_{m(l-1)}$ a tree which, by induction, is a forcing condition which is a subtree of T_τ where τ is last string in our uniform enumeration of strings of length l and has root at least as long as the height of V so far.

Now we define the U^l_k for $k \in \{0, \ldots, m(l+1)\}$. Starting with the least string $\sigma^\frown\alpha$ of length l we set $U^l_0 = R((U^{l-1}_{m(l-1)})^{V(\sigma^\frown\alpha)}_{0^{l-1}\frown\alpha}, l)$; then given U^l_k and the $k+1$th string $\sigma^\frown\alpha$ we define

$$U^l_{k+1} = R((U^l_{k+1})^{S(\sigma^\frown\alpha)}, l).$$

At the end we have $U^l_{m(l)}$, and by a similar argument as in the base case, if $\sigma^\frown\alpha$ is the kth string of length l, then $(U^l_{m(l)})^{U^l_k(\emptyset)}$ is a subtree of U^l_k which forces φ_l. So, we define

$$V(\sigma^\frown\alpha) = (U^l_{m(l)})^{U^l_k(\emptyset)}(\emptyset).$$

This preserves that V is (so far) branch coding free, congruence-preserving, and uniform, as each U^l is. We also have that the root of $U^l_{m(l)}$ is sufficiently long and has the various other properties assumed by the induction.

This completes the inductive construction of V, which is a forcing condition extending T doing no more root coding. V forces each φ_i as for every string σ of length i every path on V extending $V(\sigma)$ is a path on one of the U^i_ks, and so, by construction of the Us and the fact that the formulas are Σ^1_1, every path on V makes φ_i true, and so φ_i is forced by V. □

With the fusion lemma in hand, we can complete the proofs of the standard

lemmas regarding the forcing relation.

Proof of the density property. Let φ be a sentence in $\mathcal{L}(\omega_1^x, \mathtt{G})$ and T a forcing condition. If φ is unranked, then there is an $S \leq_{\mathcal{P}} T$ deciding φ, by the definition for forcing the negation of an unranked formula. By choosing the x-safe version of S, we can also ensure that S does no more root coding than T.

If φ is ranked, then we proceed by induction on the full ordinal rank and logical complexity of the formula. To decide atomic sentences, we can pick T_σ for some sufficiently long σ. Note that we can choose σ so as to do no more root coding.

The induction step for $\wedge$ and $\neg$ are standard; the difficulty comes with existential quantifiers. For instance, if $\varphi = (\exists X^\delta)\psi(X^\delta)$, then let $\mathcal{H}_i(n)$ be an effective enumeration of all formulas of rank at most δ whose sole free variable is n. If there is an i and an $S \leq_{\mathcal{P}} T$ such that $S \Vdash \psi(\hat{n}\mathcal{H}_i(n))$, then $S \Vdash \psi$, and we could choose the x-safe version of S so as to do no more root coding.

Otherwise, for each $S \leq_{\mathcal{P}} T$ and i, $S \nVdash \varphi(\hat{n}\mathcal{H}(n))$, and so, by induction, for each i and $S \leq_{\mathcal{P}} T$ there is an $R \leq_{\mathcal{P}} S$ such that $R \Vdash \neg\psi(\hat{n}\mathcal{H}(n))$.

Now we can apply the fusion lemma to the sequence $\neg\psi(\hat{n}\mathcal{H}_i(n))$ to find an $S \leq_{\mathcal{P}} T$ forcing each $\neg\psi(\hat{n}\mathcal{H}_i(n))$, and so $S \Vdash \neg\psi$. Furthermore, we can choose S to do no more coding, as the fusion lemma allows this.

The case for a natural number existential is similar. Observe that each extension could be chosen to do no more root coding. □

Proof that truth is forcing. Suppose $\{T_i : i \in \omega\}$ is x-generic and $\mathcal{G}$ is the generic object. Firstly, let φ be a ranked formula. Then, as the sequence is x-generic, there

is an i such that T_i decides φ. By the definition of forcing for ranked formulas, either every branch of T_i satisfies φ or every branch satisfies its negation. In particular, as $\mathcal{G} \in [T_i]$, if $\mathcal{G}^x$ satisfies φ, then every branch does, and if $\mathcal{G}^x$ satisfies its negation, then every branch does.

If φ is unranked, then we proceed by induction on the logical complexity of φ. The proof, from here, is standard. □

Lemma 4.19. *Let $\mathcal{G}$ be x-generic. Then $\mathcal{G}^x$ preserves ω_1^{CK}.*

Proof. As in Chapter IV Section 5 of Sacks [19], the fusion lemma provides the proof. □

Observe that we can construct a generic sequence by, step-by-step, extending the current condition to decide the next sentence and at no point do we have to increase the root coding of the current condition. Now we turn to showing we can force our embedding to have the properties we need.

4.4 Producing almost initial segments

We want to verify that we can construct a generic sequence $\{T_i : i \in \omega\}$ such that the generic object $\mathcal{G}$ induces an embedding $x \mapsto \text{degree}(\mathcal{G}^x)$ which preserves $\sqsubseteq$, $\not\sqsubseteq$, $\sqcup, \sqcap$, sends $\mathscr{L} \setminus \{\top\}$ to an initial segment, and sends $\top$ to $\mathcal{O}$.

Notation. For each $x \in \mathscr{L} \setminus \{\top\}$, we number the ranked terms $\mathcal{H}(t)$ of $\mathcal{L}(\omega_1^x, \mathtt{G})$ by ordinals $\delta < \omega_1^{\text{CK}}$ and denote the characteristic function of the set they stand for by $\{\delta\}^{\mathcal{G}^x}$ (of course, $\{\delta\}^{\mathcal{G}^x}$ depends on a choice of forcing condition and need not be total).

Definition 4.20. A condition T **decides** $\{\delta\}^{\mathcal{G}^x}$ **via** q (a map into $\{0,1\}$), if for every n and $\sigma \in \operatorname{dom} T$ of length l, $T_\sigma \Vdash \{\delta\}^{\mathcal{G}^x}(\underline{n}) = \underline{q(n,\sigma)}$. We say T **decides** $\{\delta\}^{\mathcal{G}^x}$ if it decides it for some q.

Lemma 4.21. *Let T be a condition which decides $\{\delta\}^{\mathcal{G}^x}$ via q. Then q is hyperarithmetic.*

Proof. The forcing relation is uniformly Π^1_1 (as the formulas are ranked), and so q is a total Π^1_1 function. Consequently, q is Δ^1_1. □

Lemma 4.22. *Let T be a condition, $x \in \mathscr{L}$, and $\delta < \omega_1^{CK}$. Then there is an $S \leq_{\mathcal{P}} T$ which decides $\{\delta\}^{\mathcal{G}^x}$, and does no more root coding than T.*

Proof. This follows from the coding-free fusion lemma. □

As our lattice representation $\{\Theta_i : i \in \omega\}$ is recursive and each Θ_i is a USL table, our map, $x \mapsto \text{degree}(\mathcal{G}^x)$, preserves $\sqsubseteq$ and $\sqcup$. We also need our map to be injective and to preserve $\not\sqsubseteq$. Injectivity follows from preservation of $\not\sqsubseteq$, so we concentrate on preserving $\not\sqsubseteq$.

We want to show for each $x \not\sqsubseteq y$ that $\mathcal{G}^x$ is not hyperarithmetic in $\mathcal{G}^y$. As $\mathcal{G}^y$ preserves ω_1^{CK} for each $y \neq \top$, it suffices to show that $\mathcal{G}_x \notin \mathcal{M}(\omega_1^y, G)$, i.e., that there is no term $\{\delta\}^{\mathcal{G}^y}$ in the language $\mathcal{L}(\omega_1^y, \mathtt{G})$ which defines $\mathcal{G}^x$. Clearly, we need not consider $y = \top$ as there is no corresponding x not below $\top$.

Lemma 4.23 (Diagonalization)**.** *Let $x, y \in \mathscr{L}$ satisfy $x \not\sqsubseteq_{\mathscr{L}} y$, let $\delta < \omega_1^{CK}$, and let T be a condition. Then there is an $n \in \omega$ and an extension of T which does no more root coding than T, decides the values of $\mathcal{G}^x(n)$ and of $\{\delta\}^{\mathcal{G}^y}(n)$, and decides them to be different.*

Proof. Firstly, as y is not above x, $y \neq \top$, so $\mathcal{L}(\omega_1^y, \mathtt{G})$ and $\mathcal{M}(\omega_1^y, G)$ are defined. We may assume T decides $\{\delta\}^{\mathcal{G}^y}$ via q (possibly by replacing T with an extension doing no more root coding), and we fix $\alpha, \beta \in \Theta_0 \setminus C$ differentiating x and y, i.e., $\alpha \equiv_y \beta$ yet $\alpha \not\equiv_x \beta$.

By the uniformity of T, there is a string π such that $T(\alpha) \supseteq T(\emptyset)^\frown\pi^\frown\alpha$ and $T(\beta) \supseteq T(\emptyset)^\frown\pi^\frown\beta$. In particular, if $n = |T(\emptyset)^\frown\pi|$, then $T_\alpha(\emptyset)(n) = \alpha$ and $T_\beta(\emptyset)(n) = \beta$. Consequently, every branch of T_α disagrees modulo x with every branch of T_β at n.

Let α^n be α concatenated with itself n times, and define β^n similarly. As T decides $\{\delta\}^{\mathcal{G}^y}$ via q,

$$T_{\alpha^n} \Vdash \{\delta\}^{\mathcal{G}^y}(n) = q(n, \alpha^n) \text{ and } T_{\beta^n} \Vdash \{\delta\}^{\mathcal{G}^y}(n) = q(n, \beta^n).$$

I claim, further, that $q(n, \alpha^n) = q(n, \beta^n)$. To see this, note that as $\alpha \equiv_y \beta$, then T_{α^n} and T_{β^n} have the same branches modulo y. This implies, by Lemma 4.17, that T_{α^n} and T_{β^n} force precisely the same sentences of $\mathcal{L}(\omega_1^y, \mathtt{G})$, which establishes the claim.

In summary, T_{α^n} and T_{β^n} force the same value of $\{\delta\}^{\mathcal{G}^y}(n)$ yet force different values of $\mathcal{G}^x(n)$. As such, at least one of T_{α^n} or T_{β^n} diagonalizes against the δth reduction. Neither tree does more root coding than T, as T is branch coding free and neither α nor β are in C, hence we have established the existence of the desired condition. □

So, by repeatedly applying the above Lemma, we can force our map to be an uppersemilattice embedding. We could force the preservation of meets by meeting appropriate dense sets, but we get it for free provided we can force the embedding to be an almost initial segment. To this end, we need to establish the existence of

splitting subtrees, which is considerably more complicated than anything we have done so far.

Definition 4.24. For each reduction δ and condition T deciding $\{\delta\}^{\mathcal{G}^x}$ via q, we say that σ and τ (of the same length) are (δ, x)**-splitting on** T **(modulo** y**)** if ($\sigma \equiv_y \tau$ and) there is an $n \leq |\sigma|$ such that $q(n, \sigma \upharpoonright n) \neq q(n, \tau \upharpoonright n)$.

Lemma 4.25. *Let δ be a reduction and T a condition deciding $\{\delta\}^{\mathcal{G}^x}$ (where $x \neq \top$). There is a ρ such that the set*

$$Sp(\rho) = \{y \in \mathscr{L} : \textit{there are no } \sigma, \tau \textit{ that } (\delta, x)\textit{-split on } T_\rho \textit{ modulo } y\}$$

is maximal. Moreover, this set is closed under meet, and so has a least element, and we can choose such a ρ which does no coding.

Proof. As $\mathscr{L}$ is finite there is clearly a ρ such that $\mathrm{Sp}(\rho)$ is maximal. I claim that if we replace ρ by its x-safe version, then $\mathrm{Sp}(\rho_x)$ is still maximal. To see this, it suffices to observe that, as T_ρ and T_{ρ_x} have the same branches modulo x, then they both decide $\{\delta\}^{\mathcal{G}^x}$ via the same map q. Hence, σ, τ (δ, x)-split on T_ρ iff they (δ, x)-split on T_{ρ_x}. Thus, $\mathrm{Sp}(\rho) = \mathrm{Sp}(\rho_x)$, and one is maximal iff the other is.

Now we need to show $\mathrm{Sp}(\rho)$ is closed under meet. Suppose $y, z \in \mathrm{Sp}(\rho)$, we want to show that there are no (δ, x)-splits on $T_{\rho^\frown 0}$ modulo $y \sqcap z$ (here we extend ρ by one place for technical reasons). Suppose there was such a split σ and τ. By the existence of meet-interpolants there are $\hat{\gamma}_1, \hat{\gamma}_2, \hat{\gamma}_3 \in \prod_{i=1}^{|\sigma|+1} \Theta_i$ such that for each $0 < j < |\sigma| + 1$ $\hat{\gamma}_1(j), \hat{\gamma}_2(j), \hat{\gamma}_3(j)$ are meet interpolants for $\sigma(j)$ and $\tau(j)$. In particular,

$$\sigma \equiv_y \hat{\gamma}_1 \equiv_z \hat{\gamma}_2 \equiv_y \hat{\gamma}_3 \equiv_z \tau.$$

Now as σ and τ form a (δ, x)-split on $T_{\rho^\frown 0}$, then so too do one of the consecutive pairs listed above. But then, supposing it is the first pair, $0^\frown\sigma$ and $0^\frown\hat{\gamma}_1$ forms a

(δ, x)-split modulo y, contradicting the fact that $y \in \mathrm{Sp}(\rho)$. The other pairs are similar, and so there are no $y \sqcap z$ splits on T_ρ as required. $\square$

Using the above lemma we construct the splitting subtrees:

Lemma 4.26. *Let δ be a reduction, T a condition deciding $\{\delta\}^{\mathcal{G}^x}$ (where $x \neq \top$) via q, ρ be a string such that $Sp(\rho)$ is maximal yet ρ does no coding, and z be the least element of $Sp(\rho)$. Then there is a condition S extending T that does no more root coding than T such that for any σ, τ if $\sigma \not\equiv_z \tau$, then σ and τ (δ, x)-split on S. We call such an S a $z - (\delta, x)$-splitting tree.*

Proof. We define S inductively, level by level. We begin with $S(\emptyset) = T(\rho)$, which, by choice of ρ, does no more root coding than T. Now suppose we have defined $S(\sigma) = T(\tau_\sigma)$ for each σ of length l. We must define $S(\sigma^\frown\alpha)$ for all such σ and $\alpha \in \Theta_l$ in a congruence respecting, branch coding free, and uniform fashion across this level.

List the strings of length $l+1$ as $\sigma_j{}^\frown\alpha_j$ for $j < m = \left|\prod_{i=0}^{l} \Theta_i\right|$. We define by a subinduction on $r < m(m-1)/2$ strings $\rho_{j,r}$ (simultaneously for $j < m$) and we will set

$$\tau_{\sigma_j{}^\frown\alpha_j} = \tau_{\sigma_j}{}^\frown\alpha_j{}^\frown\rho_{j,0}{}^\frown\cdots{}^\frown\rho_{j,m(m+1)/2}.$$

We maintain uniformity by ensuring that $|\rho_{j,r}| = |\rho_{j',r}|$ for each j, j'; we respect congruences by insisting if $\alpha_j \equiv_y \alpha_{j'}$, then $\rho_{j,r} \equiv_y \rho_{j',r}$ for each r and y; and we do no unnecessary coding by ensuring that each $\rho_{j,r}$ never takes on a coding value. Provided this is all effective, we will have a condition at the end.

By induction on $r < m(m+1)/2$ suppose we have $\tau_j{}^\frown\alpha_j{}^\frown\rho_{j,0}{}^\frown\cdots{}^\frown\rho_{j,r-1} = \nu_j$ for all $j < m$. Suppose $\{p, q\}$ is the pair of distinct numbers both less than m

numbered by r. We wish to force a split corresponding to α_p and α_q if necessary. If $\alpha_p \equiv_z \alpha_q$, then we need not force a split, and so we can define $\rho_{j,r} = \emptyset$ for each $j < m$.

Otherwise, let y be the largest $w \in \mathscr{L}$ such that $\alpha_p \equiv_w \alpha_q$, of course, $z \not\sqsubseteq y$. By choice of z, there are σ, τ such that ν_p extended by σ and τ, respectively, form a (δ, x)-splitting modulo y on T_ρ. Consequently, $\nu_q{}^\frown\tau$ must also (δ, x) split with one of $\nu_p{}^\frown\alpha$ and $\nu_p{}^\frown\tau$. If it splits with $\nu_p{}^\frown\tau$, then we set $\rho_{j,r+1} = \tau_x$ the x-safe version of τ. This is uniform and congruence respecting (because we are picking the same extension for each j) and, furthermore, $\nu_p{}^\frown\tau_x$ and $\nu_q{}^\frown\tau_x$ still form a (δ, x)-split, because $\nu_p{}^\frown\tau_x \equiv_x \nu_p{}^\frown\tau$, and so they force the same values for $\{\delta\}^{\mathcal{G}^x}$ wherever defined.

Now suppose $\nu_q{}^\frown\tau$ splits with $\nu_p{}^\frown\sigma$. If $\alpha_p \equiv_w \alpha_q$, then $w \sqsubseteq y$ by maximality of y, and so $\sigma \equiv_w \tau$, as $\sigma \equiv_y \tau$. We pick homogeneity interpolants $\gamma_0(s), \gamma_1(s)$ in Θ_{s+1} and $\mathscr{L}$-homomorphisms $f_s, g_s, h_s : \Theta_s \to \Theta_{s+1}$ such that

$$f_s : \alpha_p, \alpha_q \mapsto \sigma(s), \gamma_1(s), \quad g_s : \alpha_p, \alpha_q \mapsto \gamma_0(s), \gamma_1(s), \quad h_s : \alpha_p, \alpha_q \mapsto \gamma_0(s), \tau(s).$$

As $\nu_p{}^\frown\sigma$ and $\nu_q{}^\frown\tau$ (δ, x)-split on T_ρ, one of the pairs $\nu_p{}^\frown\sigma, \nu_q{}^\frown\hat{\gamma}_1$, or $\nu_p{}^\frown\hat{\gamma}_0, \nu_q{}^\frown\hat{\gamma}_1$, or $\nu_p{}^\frown\hat{\gamma}_0, \nu_q{}^\frown\tau$ must (δ, x)-split on T_ρ too. We set $\rho_{j,r+1}(s) = f_s(\alpha_j)$ or $g_s(\alpha_j)$ or $h_s(\alpha_j)$ corresponding to which pair splits. This is uniform as $\rho_{j,r+1}$ depends only on α_j, and, as each f_s, g_s, h_s are $\mathscr{L}$-homomorphisms, we respect congruences.

The final thing to show is that we have not done unnecessary coding. Well, as $\{\Theta_i : i \in \omega\}$ is coding ready, then it is acceptable for A, so, by definition, $f_s(\alpha) \in C$ implies that $\alpha = \sigma(s)$ or $\alpha = \tau(s)$ (and the same for g_s, h_s). So, it suffices to show that we can pick σ and τ which themselves do no coding.

As $\nu_p = \tau_{\sigma_j}{}^\frown\alpha_j{}^\frown\rho_{j,0}{}^\frown\cdots{}^\frown\rho_{j,r-1}$, then $|\nu_p| > 0$; therefore, the σ, τ we are

trying to pick live in $\prod_{i=k}^{i=k'} \Theta_i$ for some $k' > k > 0$. Consequently, if $\sigma(s) \in C$, then $\sigma(s) \in \Theta_0 \subseteq \Theta^*_{k+s}$ (similarly for $\tau(s)$), and so, by the definition of acceptable for A, there are $\sigma'(s)$ and $\tau'(s) \in \Theta_{k+s} \setminus \Theta^*_{k+s}$ such that

$$\sigma(s) \equiv_{x'} \sigma'(s), \tau(s) \equiv_{x'} \tau'(s), \text{ and for all } w \in \mathscr{L}[\sigma(s) \equiv_w \tau_s \Rightarrow \sigma'(s) \equiv_w \tau'(s)].$$

If we pick a coatom $x' > x$, then we can construct σ', τ' which do not take coding values and such that $\sigma' \equiv_{x'} \sigma$ and $\tau' \equiv_{x'} \tau$, and, therefore, which still form a (δ, x)-split on T_ρ modulo y. Thus, when we picked σ, τ we could have picked them to do no coding, and then nothing else can code as $\{\Theta_i : i \in \omega\}$ is acceptable for A. □

Lemma 4.27. *If T is a $z - (\delta, x)$-splitting tree, then T forces $\{\delta\}^{\mathcal{G}^x} \equiv_h \mathcal{G}^z$.*

Proof. Fix $\mathcal{G} \in [T]$. We first show $\mathcal{G}^z \geq_h \{\delta\}^{\mathcal{G}^x}$. Pick an n. Using $\mathcal{G}^z$ find all $\sigma \in \operatorname{dom} T$ of length n such that $T(\sigma)(m)(z) \equiv_z \mathcal{G}^z(m)$ for all $m \leq n$, i.e, narrow down the possible paths through T to those consistent with $\mathcal{G}^z$. All these σ are equivalent modulo z, and so each T_σ forces the same value of $\{\delta\}^{\mathcal{G}^x}(n)$, by the choice of z. As $T(\sigma)$ is an initial segment of $\mathcal{G}$ for one of these σ, then $\{\delta\}^{\mathcal{G}^x}(n)$ must be the correct value for that σ, and hence, all such σ.

Now we compute the other way. Given $\{\delta\}^{\mathcal{G}^x}$ consider all $\sigma, \tau \in \operatorname{dom} T$ of length n. If $\sigma \not\equiv_z \tau$, then σ and τ form a (δ, x)-split on T and so, in particular, T_σ and T_τ force different values for $\{\delta\}^{\mathcal{G}^x}$ at some $m < n$. Thus, of the $\equiv_z$ equivalence classes of σs and τs of length n, only the correct one will force the correct value of $\{\delta\}^{\mathcal{G}^x}$, so we can rule out all of the incorrect one, as T and q are hyperarithmetic. This leaves us with a single $\equiv_z$ equivalence class, and every member determines the same initial segment of $\mathcal{G}$ modulo z, and so the initial segment must be correct modulo z. □

This is the penultimate step on our way to our lattice embedding theorem: We know we can construct an embedding of $\mathscr{L}$ into $\mathcal{D}_h$ such that $\mathscr{L} \setminus \{\top\}$ is an initial segment. We do this by diagonalizing against each hyperarithmetic reduction δ for each pair $x \not\sqsubseteq y \in \mathscr{L}$ and by constructing splitting trees. This is only countably many requirements so we can satisfy each in turn. We also need to decide each sentence of $\mathcal{L}(\omega_1^x, \mathtt{G})$ for each $x \in \mathscr{L} \setminus \{\top\}$ and to preserve ω_1^{CK} for each such x.

We have proved that we can do all this without ever doing coding at the root of a condition, consequently, we can intersperse these requirements with requirements saying:

If n is the first place we are yet to code for the pair x, y joining up nontrivially to $\top$, then if $n \in \mathcal{O}$ take $T_{g(x,y,1)}$ and if $n \notin \mathcal{O}$ take $T_{g(x,y,0)}$ as the next condition (where g is the function in the definition of coding set).

This implements our coding scheme, and so $\mathcal{G}^\top \geq_h \mathcal{O}$. How do we guarantee that $\mathcal{G}^\top \leq_h \mathcal{O}$? It suffices to construct a generic sequence hyperarithmetically in $\mathcal{O}$. But this is only so much checking: The notion of forcing and the extension relation are hyperarithmetic in $\mathcal{O}$, as are the languages $\mathcal{L}(\omega_1^x, \mathtt{G})$. Also, the various constructions we effected can all be made uniformly hyperarithmetic in $\mathcal{O}$ by always picking "least" strings or extensions doing various things, with some fixed hyperarithmetic enumeration of strings.

The reader may also note that we could code in any set for $\top$, and provided that set X is hyperarithmetically above $\mathcal{O}$ then there is a generic $\mathcal{G}$ with top element having the same hyperdegree as X. Thus we have produced the required almost initial segments, and so answered the extension of embeddings problem for finite USLs and USL$^\top$s into $\mathcal{D}_h$ and $\mathcal{D}_h(\leq_h \mathcal{O})$ respectively.

4.5 Remarks

The fact that every finite lattice embeds as an initial segment of $\mathcal{D}_h$ suffices to show that the Σ_3 theory of $\mathcal{D}_h$ as a poset is undecidable (see Lerman [15] VII.4.6). This also applies to $\mathcal{D}_h(\leq_h \mathcal{O})$. Consequently, we are on the edge of decidability with our current results. Further questions would involve adding jump or meet to the language (as not every pair of hyperdegrees has a meet, some technical modifications must be made).

As for countable initial segments, it is not true that every countable lattice is an initial segment of $\mathcal{D}_h$ (This is a Theorem of Shore, see [11] Theorem 1.1). However, Kjos-Hanssen and Shore [11] have shown that every sublattice of every hyperarithmetic lattice can be embedded as an initial segment with $\top$ mapping to a degree below the join of $\mathcal{O}$ and the lattice itself.

To embed a countable lattice as an initial segment as above, one cannot just start with a USL table for the whole lattice, as then the trees would be infinitely branching and so not amenable to perfect forcing. Kjos-Hanssen and Shore circumvent this problem by approximating the lattice by a sequence of nested finite substructures, and having the USL tables at a finite stage only represent one of these growing finite substructures. The author believes that these methods could be adapted in full to show that if $\mathscr{L}$ is a sublattice of a hyperarithmetic lattice, then $\mathscr{L}$ embeds as an almost initial segment of $\mathcal{D}_h(\leq_h \mathcal{O} \oplus \mathscr{L})$.

CHAPTER 5

A THEOREM OF HYPERARITHMETIC ANALYSIS

Now we turn to something quite different: some reverse mathematics. The project of reverse mathematics attempts to determine the strength of theorems in terms of what axioms are necessary and sufficient to prove them. The main reference for the topic of reverse mathematics is Simpson's *Subsystems of second order arithmetic*[23]. In this chapter we present some preliminary results regarding the strength of a result in graph theory. This is joint work with Richard Shore and Jun Le Goh.

5.1 Some reverse math

We start with the second-order language of arithmetic $\mathcal{L}_2$ generated from the following nonlogical symbols:

- two binary function symbols $+, \times$;
- and two constant symbols $0, 1$.

Our language also has logical connectives, variables of two different types, quantifiers for the two different types, and symbols for $=$ and $\in$.

A **model** in the language $\mathcal{L}_2$ is, as normal, a collection $\mathscr{M}$ of objects with a collection $\mathscr{S}$ of subsets of $\mathscr{M}$ along with interpretations of the symbols $+, \times, 0, 1$. If $\mathscr{N} = \omega$ and $+, \times, 0, 1$ get their standard interpretations, then the model is called an ω-model. Clearly, ω-models are fully determined by their second order part, and so we will often call a collection of subsets of ω an ω-model.

Robinson arithmetic $\mathcal{Q}$ is the following collection of axioms (or rather, their universal closures are the axioms):

- $x + 1 \neq 0$,
- $x + 1 = y + 1 \to x = y$,
- $x \neq 0 \to (\exists y)[y + 1 = x]$,
- $x + 0 = x$,
- $x + (y + 1) = (x + y) + 1$,
- $x \times 0 = 0$,
- $x \times (y + 1) = x \times y + x$.

. The (set) **induction axiom** is the (universal closure of)

$$(0 \in X \wedge (\forall x)[x \in X \to x + 1 \in X]) \to (\forall x)[x \in X].$$

For a formula $\varphi(x)$ of $\mathcal{L}_2$ the **induction axiom for** φ is the formula

$$(\varphi(0) \wedge (\forall x)[\varphi(x) \to \varphi(x + 1)]) \to (\forall x)\varphi(x).$$

If Γ is a class of formulas, then Γ-induction is the scheme of induction axioms for each $\varphi \in \Gamma$, in particular, Σ^0_1-induction is the collection of induction axioms for formulas φ that are Σ^0_1 (note that they may contain parameters and bounded quantifiers).

A **comprehension axiom** says, roughly, that if you have a description for a class of objects, then there is a set containing those objects. The Δ^0_1 comprehension scheme is the set of all axioms of the form

$$(\forall x)[\varphi(x) \leftrightarrow \psi(x)] \to (\exists X)(\forall x)[x \in X \leftrightarrow \varphi(x)],$$

where φ is Σ^0_1, ψ is Π^0_1, and X is not free in either φ or ψ.

RCA_0 is the collection of axioms containing $\mathcal{Q}$, the set induction axiom, Σ^0_1 induction, and Δ^0_1 comprehension, and ACA_0 is RCA_0 with arithmetical comprehension:

$$(\exists X)(\forall x)[x \in X \leftrightarrow \varphi(x)]$$

where φ is arithmetical (and X is not free in φ).

It is obvious that ACA_0 implies RCA_0. This implication does not reverse because the recursive sets form an ω-model of RCA_0 that is not a model of ACA_0. Indeed, the following is a key fact about ω-models of these theories:

Theorem 5.1 (See Simpson [23] VIII.1.1 and VIII.1.10)**.** *An ω-model $\mathscr{S}$ satisfies RCA_0 iff it is a nonempty **Turing ideal**, i.e., iff it is closed under $\oplus$ and downward under $\leq_T$. It satisfies ACA_0 iff it is a nonempty **jump ideal**, i.e., iff it is a Turing ideal also closed under the Turing jump.*

For a formula $\varphi(n, X)$ the **φ-choice axiom** is

$$(\forall n)(\exists X)[\varphi(n, X)] \rightarrow (\exists X)(\forall n)[\varphi(n, X^{[n]})].$$

Σ^1_1 choice is the scheme of choice axioms for Σ^1_1 formulas and Σ^1_1-AC_0 is the system ACA_0 with the Σ^1_1 choice axiom scheme added. It is known that Σ^1_1-AC_0 strictly implies ACA_0.

Definition 5.2 (Theory of hyperarithmetic analysis)**.** A set of sentences Γ in second-order arithmetic is a **theory of hyperarithmetic analysis** if for every set X the least ω-model of Γ containing X is $\mathrm{HYP}(X)$.

Proposition 5.3 (Kreisel [14] for the unrelativised version)**.** *Σ^1_1-AC_0 is a theory of hyperarithmetic analysis.*

5.2 Some graph theory

A **graph** $G = (V, E)$ is a pair where V is a set of **vertices** and E is a set of **edges** between members of v, i.e., each $e \in E$ is of the form $\{v, w\}$ for distinct $v, w \in V$. A **path** in the graph G is a sequence of distinct vertices where consecutive members in the sequence are joined by an edge in G. We think of a path as being traversed by an agent walking between vertices along edges of the graph.

A (one-sided) infinite path in G is called a **ray**, and a path that is infinite in both directions is called a **double-ray**. More explicitly, a ray R is a function from ω to V such that $\{R(n), R(n+1)\} \in E$ for each $n \in \omega$, and a double-ray is a function from $\mathbb{Z}$ to V satisfying the same property for each $n \in \mathbb{Z}$. A family of rays $\mathcal{R}$ in G is **(vertex) disjoint** if for every distinct $R_1, R_2 \in \mathcal{R}$ the ranges of R_1, R_2 are disjoint. It is **(edge) disjoint** if for each distinct $R_1, R_2 \in \mathcal{R}$ there are no $n, m \in \omega$ such that $\{R_1(n), R_1(n+1)\} = \{R_2(m), R_2(m+1)\}$, i.e., the rays R_1, R_2 don't share any edges. There are similar definitions for double rays. When we say rays are disjoint, we mean vertex disjoint.

The theorem we are interested in we call the infinite rays theorem:

Theorem 5.4 (Originally Halin[8], see Diestel [5] 8.2.5)**.** *If G is an infinite graph that contains a family of k disjoint rays for each $k \in \omega$, then G contains an infinite family of disjoint rays.*

Proof. We construct the rays by induction.

Stage 1: Pick a ray R_1^1 in G, and let P_1^1 be the path consisting of the first vertex of R_1^1.

Stage $n+1$: By induction we have n many disjoint rays $R_1^n, \ldots, R_n^n$ and n

many paths $P_1^n, \ldots, P_n^n$ where P_i^n is an initial segment of R_i^n. At the end of the stage we will have extended each of the paths by a finite amount and started a new path; everything will be kept disjoint. We will maintain that these paths can be extended to disjoint rays.

Start by picking a family $\mathcal{R}$ of $n^2 + 1 + |\bigcup_{i=1}^n P_i^n|$ many disjoint rays. Remove from $\mathcal{R}$ any of the rays that intersect any of the finite paths P_i^n for $i = 1, \ldots, n$. As the rays in $\mathcal{R}$ are disjoint, then we have removed at most $|\bigcup_{i=1}^n P_i^n|$ many rays, and so there are at least $n^2 + 1$ many members of $\mathcal{R}$ remaining none of which intersect any of the P_i^ns.

Repeat the following step as many times as possible: if there exists an $i \in \{1, \ldots, n\}$ such that R_i^{n+1} has not been defined and R_i^n meets at most n of the members of $\mathcal{R}$, then define $R_i^{n+1} = R_i^n$, delete from $\mathcal{R}$ any members meeting R_i^{n+1}, and define P_i^{n+1} to be the initial segment of R_i^{n+1} containing one more vertex than P_i^n.

Let I be the subset of $\{1, \ldots, n\}$ for which we did not act in the last step. Then $\mathcal{R}$ contains at least $n^2 + 1 - (n - |I|)n = 1 + n|I| \geq 1 + |I|^2$ many rays, as we acted $n - |I|$ many times, and each time we acted we removed at most n many rays.

Each ray R_i^n with $i \in I$ meets at least $n \geq |I|$ many members of $\mathcal{R}$. Let z_i be the first vertex of the $|I|$th ray it meets and let Z equal the union of the paths between the last vertex on P_i^n and z_i along R_i^n for $i \in I$. The subgraph Z meets at most $|I|^2$ many members of $\mathcal{R}$ (as each path meets $|I|$ many and there are $|I|$ many paths). Delete every other ray in $\mathcal{R}$, choosing one of them to be R_{n+1}^{n+1} and P_{n+1}^{n+1} is the first vertex of R_{n+1}^{n+1}.

For each $R \in \mathcal{R}$ pick a vertex $y(R)$ after its last vertex in Z. Let $Y = \{y(R) :$

$r \in \mathcal{R}\}$, and let H be the union of Z with the initial segment of R up to $y(R)$ for each $R \in \mathcal{R}$. Let X be the set of final vertices in P_i^n for $i \in I$. Observe X and Y are disjoint subsets of H. Furthermore, if you delete vertices in H until there is no path in H between any $x \in X$ and $y \in Y$, then you would have to delete at least $|I|$ many vertices because deleting $|I| - 1$ many vertices would mean there exists $i \in I$ such that you deleted no part of the path between the last vertex on P_i^n and z_i (by disjointness of these over $i \in I$), and also there would remain at least one initial segment of some $R \in \mathcal{R}$ meeting the path corresponding to i because there are $|I|$ many such paths and they are all disjoint.

Consequently, by Menger's Theorem, there are $|I|$ many mutually disjoint paths in H which join members of X to members of Y. Consequently the paths that start with P_i^n, then continue along the path joining the corresponding $x \in X$ to some $y \in Y$, and then continue along the corresponding ray $R \in \mathcal{R}$ are all mutually disjoint. Let R_i^{n+1} equal the ray just described, and extend P_i^{n+1} by one step in the new ray. This completes the $n + 1$st step.

Our family of rays are defined by defining $Q_n = \lim_{m\to\infty} P_n^m$. This is well defined as P_n^{m+1} is a proper end extension of P_n^m for all $m \geq n$. □

There are versions of this theorem for edge disjoint rays, as well as vertex and edge disjoint versions for double rays, but we do not consider them here.

5.3 The strength of the infinite ray theorem

There are a number of different noncomputable steps in Diestel's proof. Firstly we must pick a family $\mathcal{R}$ of rays. Secondly we must detect which rays intersect other

rays or finite paths. Everything else (including Menger's Theorem) is uniformly computable. Given two rays detecting whether they intersect is computable in the jump of their join and given a ray and a finite set, detecting whether the ray intersects the finite set is computable in the jump of the ray. Consequently ACA_0 suffices to effect steps of the second kind.

The first noneffective step is more complicated, and it depends on how we choose to formalize the hypothesis of the theorem. To wit:

- Is there one set the kth column of which contains k disjoint rays?
- Is there one set such that each column is a ray, and for each k there are k many columns that are disjoint, but, as it were, we "don't know where to look"?
- Do we not assume that we have one set doing all the work, just that for each k there is a set of k many disjoint rays?

These hypotheses decrease in strength, and so the theorem increases in strength the weaker the hypothesis you assume. These lead to our formalized theorems:

Definition 5.5 (IRT and variants)**.** (Strong) IRT is the following principle:

> If $G = (V, E)$ is a graph and for each k there is a set X_k such that $X_k^{[i]}$ is a ray in G for $i = 0, \ldots, k-1$ and $X_k^{[i]}$ is disjoint from $X_k^{[j]}$ for $0 \leq i < j \leq k-1$, then there is a set Y such that $Y^{[i]}$ is a ray in G for each $i \in \omega$ and $Y^{[i]}$ is disjoint from $Y^{[j]}$ for $0 \leq i < j < \omega$.

Nonuniform-WIRT is the principle IRT but with hypothesis on G being replaced with:

> There is a set X each column of which is a ray in G, and for each k there are columns $X^{[i_0]}, \ldots, X^{[i_{k-1}]}$ that are pairwise disjoint.

Finally, uniform-WIRT is the principle IRT with the hypothesis in G being replaced with:

> There is a set X such that for each k, the first k columns of $X^{[k]}$ are a collection of pairwise disjoint rays in G.

Recall that a we are thinking of a ray as a function from ω to V, so saying "$X_k^{[i]}$ is a ray" abbreviates "$X_k^{[i]}$ is a function with domain ω and codomain V, such that $\{X_k^{[i]}(n), X_k^{[i]}(n+1)\} \in E$ for each n" and saying that two rays are disjoint abbreviates that their images are disjoint in V.

Now we turn to the various implication and nonimplication results we have determined involving the variants of the infinite ray theorem.

Proposition 5.6. *In* RCA_0 *it is provable that* IRT *implies nonuniform-*WIRT*, which in turn implies uniform-*WIRT*.*

Proof. Arguing in RCA_0, it is clear that the uniform hypothesis implies the nonuniform hypothesis which implies the "strong" hypothesis. As the conclusion of the principles is the same in each case, the implications follow. □

Proposition 5.7. *None of the three variants of the theorem are provable in* RCA_0.

As uniform-WIRT is implied by the other two variants, it suffices to show that uniform-WIRT is not a theorem of RCA_0. In particular, it suffices to show that the recursive sets are not an ω-model of uniform-WIRT. To this end we need to produce

a computable graph G with the property that there is a computable function that takes n and returns a (procedure to compute) a family of n disjoint rays in G, but so that there is no computable infinite set of disjoint rays. To make this clear we take a quick detour.

Lemma 5.8. *There is an r.e. equivalence relation $\sim$ such that one can uniformly in n compute an $\sim$-independent set of size n, yet there is no infinite r.e. $\sim$-independent set.*

Proof. The field of $\sim$ will be $F = \{\langle n, m\rangle : n < m\}$. We call the set $B_m = \{\langle n, m\rangle : n < m\}$ the mth **block**. For each e there is a **requirement**

$$R_e : \ \Phi_e \text{ does not enumerate an infinite } \sim\text{-independent set.}$$

The requirement R_e is of **higher priority** than R_i if $e < i$. A requirement becomes **satisfied** if it has enumerated two different members of F, and we have enumerated the equivalence of those members into $\sim$, otherwise a requirement is **unsatisfied**.

A requirement R_e **restrains** an equivalence class if it is unsatisfied and Φ_e has enumerated some $\langle n, m\rangle$ such that the equivalence class has a member that shares a block with a member of the equivalence class of $\langle n, m\rangle$, i.e. for all $\langle n', m'\rangle \sim \langle n, m\rangle$ and all $i < m'$, R_e restrains the equivalence class of $\langle i, m'\rangle$. If a requirement restrains an equivalence class, then no requirement of lower priority may enumerate new members into that class.

Construction: Simultaneously enumerate Φ_e for each e. R_e requires attention if

- it is yet to be satisfied;

- it has enumerated elements $\langle n_0, m_0\rangle, \langle n_1, m_1\rangle$ such that $m_0 \neq m_1$, the equivalence class of neither element has been restrained by a requirement of higher priority, and there is no m', n_0', n_1' such that $\langle n_0, m_0\rangle \sim \langle n_0', m'\rangle$ and $\langle n_1, m_1\rangle \sim \langle n_1', m'\rangle$ have been enumerated into $\sim$ before.

Each unsatisfied requirement restrains equivalence classes as described above. As they become satisfied, they release their restraint. If a requirement requires attention, then we **act** by enumerating the equivalence $\langle n_0, m_0\rangle \sim \langle n_1, m_1\rangle$ (and the other way round for symmetry), and then closing $\sim$ under transitivity. Also, we enumerate $\langle n, m\rangle \sim \langle n, m\rangle$ for each $\langle n, m\rangle \in F$ in between stages, and if Φ_e enumerates something that is not a member of F, then we declare it satisfied so it releases all restraints.

Verification: Firstly, $\sim$ is reflexive, symmetric, and transitive. Secondly, we never act to add an equivalence between members of the same block, and we never add a member that would imply an equivalence between members of the same block using transitivity. Consequently, each block is $\sim$-independent. This shows that there are uniformly computable $\sim$-independent sets of size n.

Now we show each requirement is satisfied. Suppose R_i has been satisfied for all $i < e$. If Φ_e enumerates a pair not in F, or only finitely many elements, then Φ_e does not compute an $\sim$-independent infinite set. Otherwise Φ_e enumerates an infinite subset of F. If R_e ever requires attention, then we enumerate an equivalence into $\sim$ that means that Φ_e is not $\sim$-independent. Hence, it suffices to show that R_e requires attention at some stage.

Let $\langle n_0, m_0\rangle$ be the first pair enumerated into R_e that is not restrained by a requirement of higher priority. Such a pair exists because for each $i < e$, R_i

either becomes satisfied at some finite stage and so no longer restrains anything, or it never becomes satisfied during the construction, but it is satisfied at the end, therefore it must have only enumerated finitely many elements of F and so only restrains finitely many classes.

Then, at some later stage Φ_e enumerates some element $\langle n_1, m_1 \rangle$ which satisfies the above three conditions for $\langle n_0, m_0 \rangle$, as R_e has restrained all classes which every share a block with the class of $\langle n_0, m_0 \rangle$. When this first happens we act by enumerating the equivalence between $\langle n_0, m_0 \rangle$ and $\langle n_1, m_1 \rangle$ into $\sim$, and so R_e becomes satisfied, and so W_e is not $\sim$-independent. $\square$

Our lemma can be used to prove Proposition 5.7.

Proof of Proposition 5.7. Using the $\sim$ of Lemma 5.8. Let f be a computable enumeration of $\sim$. We build a graph $G = (V, E)$ on the product of the field of $\sim$ and ω, i.e., $V = \{\langle n, m \rangle : n < m\} \times \omega$. We will grow rays upward from $\langle n, m, 0 \rangle$ for each $n < m$, and merge the rays corresponding to things we find to be $\sim$-equivalent.

At stage 0 declare $\langle n, m, 0 \rangle$ adjacent to $\langle n, m, 1 \rangle$ for each $n < m$. At stage $s > 0$, we have defined all the adjacencies between members whose third coordinate is less than s. Compute $f(s-1)$. Given $\langle n, m, s \rangle$, check whether there are $n' < m' \leq m$ such that $\langle n', m', s-1 \rangle$ is adjacent to $\langle n, m, s \rangle$. If not, then we do not add anything to $\langle n, m, s \rangle$. Otherwise, if $f(s-1)$ is a (nonidentity) $\sim$ congruence involving $\langle n, m \rangle$ and some other $\langle n', m' \rangle$, then declare $\langle n, m, s \rangle$ adjacent to $\langle \max\{n, n'\}, \max\{m, m'\}, s+1 \rangle$. If not, then declare $\langle n, m, s \rangle$ adjacent to $\langle n, m, s+1 \rangle$. This completes stage s.

The construction tries to extend the ray starting at $\langle n, m, 0 \rangle$ upward. If $\langle n, m \rangle$

is not declared equivalent to anything, then we just increase the ray upward. If $\langle n, m\rangle$ is declared equivalent to $\langle n', m'\rangle$, then we merge the rays corresponding to these (into the one with the larger m, and as $m \neq m'$ by definition of $\sim$ this is well defined). Consequently, uniformly in n you can compute n many disjoint rays by starting at $\langle 0, n, 0\rangle, \langle 1, n, 0\rangle, \ldots, \langle n-1, n, 0\rangle$ and moving upward in the third coordinate at each step. There is always a unique way to go, and none of these ever intersect by construction. However, any infinite collection of disjoint rays can be used to compute an infinite $\sim$-independent set, which is a contradiction. □

So none of our principles are computably true. The weaker two are, however, provable in ACA_0.

Proposition 5.9. ACA_0 *implies the nonuniform-*WIRT *and consequently the uniform-*WIRT*.*

Proof. From the discussion above, Diestel's proof has two steps that are not computable: detecting when rays intersect, and picking the families of disjoint rays. Let X be the set in the hypothesis of the nonuniform$-\mathsf{WIRT}$, so each column of X is ray, and for each n there are $i_0, \ldots, i_{n-1}$ such that $X^{[i_0]}, \ldots, X^{[i_{n-1}]}$ are disjoint. ACA_0 can detect whether $X^{[i]}$ and $X^{[j]}$ intersect, and so whenever we need to find a finite set of disjoint rays ACA_0 can search through the columns of X to find a set of the right size. The hypothesis on X says this search must terminate.

Similarly at the parts of the proof were we try to detect whether new rays intersect some finite set, ACA_0 suffices to answer this question for us. The rest of the proof is uniform and computable. □

We also have a reversal.

Proposition 5.10. *Over* RCA_0*, nonuniform-*WIRT *implies* ACA_0*.*

Proof. It suffices to show that over RCA_0, nonuniform-WIRT implies that the Turing jump of every set exists. Let $\nabla(i)$ be the **true stage function**

$$\nabla(i) = (\mu s)[s > \nabla(i-1) \text{ such that } \Phi_i(i) \downarrow \leftrightarrow \Phi_{i,s}(i) \downarrow],$$

where, by convention $\nabla(-1) = -1$. Thus $\nabla(i)$ is the first stage (greater than the last true stage) in the standard enumeration of $\emptyset'$ where the enumeration is correct up to i. It is not hard to see that any function f that dominates ∇ computes $\emptyset'$. Firstly, change f on a finite set of initial values so that $f(n) \geq \nabla(n)$ for all n. Then to compute whether $n \in \emptyset'$, you compute $f(n) \geq \nabla(n)$. Compute $\Phi_{n,f(n)}(n)$. By the definition of ∇, if $\Phi_n(n)$ is ever going to halt, it would have done so in $\nabla(n) \leq f(n)$ many steps. Hence, $\Phi_{n,f(n)}(n)$ halts iff $n \in \emptyset'$.

Our goal, then, is to construct a graph that RCA proves is an instance of nonuniform-WIRT, yet any infinite family of disjoint rays can be used to construct a function dominating the true stage function. If our construction relativises, then we will be done.

Our graph G will be on ω^2. At each node $(n, 0)$ we will start by growing a ray "upward". Simultaneously, we will be enumerating $\emptyset'$. If at some stage we see that a number n is not the mth true stage (e.g. we thought that stage 10 is the 5th true stage at stage 10, and we just enumerated 7, but then at stage 20 we enumerated 4, therefore stages $10 - 19$ are not true stages), then we merge the rays between our old guess for the mth true stage and our new guess into our new guess. The figure below illustrates an instance where we learned that 0 and 1 are not true stages.

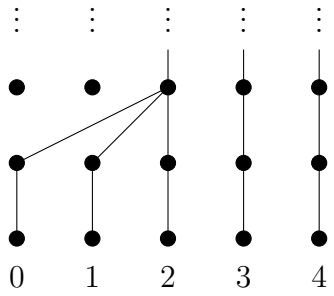

Figure 1

This can only happen because we enumerated 0 into $\emptyset'$ at stage 2. At this point we know $\nabla(0) = 2$, our best guess for $\nabla(1)$ at this point is 3. If $1 \notin \emptyset'$, then our guess is correct, but we will never know this for sure. We formulise this guessing procedure by a family of partial functions: for $j < t$ the j**th apparent true stage at** t is

$$\nabla_t(j) = (\mu s \leq t)[s > \nabla_t(j-1) \text{ such that } (\forall n \leq j)\Phi_{n,s}(n) \downarrow \Leftrightarrow \Phi_{n,t}(n) \downarrow].$$

Observe that ∇_t is uniformly computable in t and j (again, we assume $\nabla_t(-1) = -1$). The idea of the function is that you guess that stage t is a true stage, and then use it to find all the smaller true stages. Also note that $\lim_{t\to\infty} \nabla_t(j) = \nabla(j)$.

Construction of $G = (V, E)$*:* Our vertex set is $V = \{\langle n, m\rangle : n, m \in \omega\}$. We need to construct E and our set X of rays. Firstly, we add an edge between $\langle n, 0\rangle$ and $\langle n, 1\rangle$ for each n. We initialize the set of dead columns to be $\emptyset$.

Then at stage t, we want to determine what to join $\langle n, t\rangle$ to in the row above, if anything. Firstly, check whether n is a dead column; If it is, then we do not adjoin anything. Otherwise, n is not dead. If $t \leq n$, then add an edge to $\langle n, t+1\rangle$. Otherwise $t > n$. Observe that $\nabla_t(n) \geq n$, and $\nabla_t(j)$ is increasing in j, so find the least j such that $\nabla_t(j) = m \geq n$; add an edge between $\langle n, t\rangle$ and $\langle m, t+1\rangle$. If $m \neq n$, then declare the n-th column dead.

Observe that if a column n is not dead at stage t, then there is a unique m such that $\langle n, t\rangle$ and $\langle m, t\rangle$ are adjacent. Also, no columns start out dead. Thus starting at $\langle n, 0\rangle$ there is always a unique node in the next level up that you can walk to; this give our set of rays: $X^{[n]}(0) = \langle n, 0\rangle$ and if $X^{[n]}(x) = \langle m, x\rangle$, then $X^{[n]}(x+1) = \langle l, x+1\rangle$ for the unique l such that $\langle m, x\rangle$ and $\langle l, x+1\rangle$ are adjacent.

Verification: As $\nabla_t(j)$ is uniformly computable, then both G and X are computable, and so RCA_0 proves they exist by Δ^0_1 comprehension. Each column of X is a ray, and further, RCA_0 proves that for every n there are n many true stages, and so proves that the corresponding columns of X are disjoint. Thus, we have an instance of nonuniform-WIRT.

Let Y be an infinite family of disjoint rays in G. Any ray always moves "upward" in G. For each true stage $\nabla(n)$ denote the interval $(\nabla(n), \nabla(n+1)]$ the nth block (starting at $n = -1$). All of the upward growing rays in G eventually merge to the first true stage to their right, i.e., to the right endpoint of the block in which they start. Consequently, by disjointness of the rays in Y, at most one ray has any node in any block, and each ray remains entirely in the vertical strip defined by the block in which it starts.

So, we want to dominate the true stage function. To compute a value greater than $\nabla(0)$ examine $Y^{[0]}(0)$ and $Y^{[1]}(0)$. By the above, these two nodes live in vertical strips defined by different blocks. As such, at least one of them lives in a block with left end point $\nabla(i)$ for $i \geq 0$. Therefore define $f(0)$ to be the larger of the first coordinates of $Y^{[0]}(0)$ and $Y^{[1]}(0)$. Then, similarly, define $f(n)$ to be the largest of the first coordinates of $Y^{[0]}(0), \ldots, Y^{[n]}(0)$. The function f dominates the true stage function, and so it computes $\emptyset'$. As f is computable from Y, we have constructed $\emptyset'$. To relativise this to a set X, you use the true stage function for

X' which is X-computable. □

We have not succeeded in separating the uniform and nonuniform WIRTs. Now we turn to IRT.

Proposition 5.11. Σ^1_1-AC_0 *implies* IRT.

Proof. Firstly, recall that Σ^1_1-AC_0 implies ACA_0. Let $\varphi(n, X)$ be the formula that says "the first n columns of X are disjoint rays in G." This formula is Σ^1_1, and by the hypothesis of IRT, for every n there is an X satisfying the formula (provided G satisfies the hypothesis). Consequently using the instance of choice for this formula, we can construct a set $\tilde{X}$ the nth column of which is a set of n many disjoint rays, i.e., $\tilde{X}$ satisfies the hypothesis of the uniform-WIRT.

Thus as Σ^1_1-AC_0 implies ACA_0 and we have transformed our instance of IRT into an instance of uniform-WIRT, then by proposition 5.9, Σ^1_1-AC_0 produces an infinite set of disjoint rays in G. □

Theorem 5.12. IRT *is a theory of hyperarithmetic analysis.*

Proof. Firstly, as Σ^1_1-AC_0 implies IRT, then every model of Σ^1_1-AC_0 is a model of IRT. In particular, for each set Z, HYP(Z) is a model of IRT. Now we want to show that every ω-model of IRT containing Z contains HYP(Z).

As IRT implies the nonuniform-WIRT which implies ACA_0, then every ω-model of IRT is an ω-model of ACA_0 and so is closed under Turing jump. Consequently, we need to show that we can prove the existence of limit level jumps. It is a fact (see Chong and Yu [4] Section 2.1.2 for proofs, in particular Theorem 2.1.4 and Proposition 2.1.5) that every H-set H_a is a Π^0_2**-singleton**, i.e., H_a is the unique

solution to some Π^0_2 subset of 2^ω. Furthermore, every Π^0_2 singleton is the unique branch on a recursive subtree of ω^ω. Finally, the Π^0_2 predicate defining H_a, and the recursive tree corresponding to the Π^0_2 predicate are all determined uniformly.

Consequently, suppose α is a computable limit ordinal so $\alpha = |3 \cdot 5^e|$. By induction, assume that $H_{\Phi_e(n)}$ is in our model for each n. Let T_n be a computable subtree of ω^ω whose unique path is $H_{\Phi_e(n)}$. This is uniform in n, so let G be the disjoint union of the T_n, which is also computable and so in the model.

G is a union of countably many trees all with unique paths in our model. Consequently, for each n there is a set of n many of the paths, which are n disjoint rays in G. Therefore G is an instance of IRT in our model, so it has a solution Y in the model. Each column of Y is a ray in G, hence it is an infinite path in one of the trees T_n. We can uniformly in m detect which T_n the ray $Y^{[m]}$ lives in, and we can then uniformly change $Y^{[m]}$ so that it starts at the root of T_n but so that the old and new versions share tails. Consequently, we can assume that Y is a set of the unique paths starting at the root of some T_n. We can also assume that they occur in increasing order of n. Then $Y \geq_T H_{3 \cdot 5^e}$. Relativising this result to Z poses no problems. □

This leads to the next easy corollary.

Corollary 5.13. IRT *is not implied by* ACA_0*, and so* IRT *is not implied by either of the* WIRT*s.*

Proof. The arithmetical sets form an ω-model for ACA_0, yet, by Theorem 5.12, the least ω-model of IRT containing $\emptyset$ is HYP. □

In summary we can compare the strengths of our systems over RCA_0 as follows:

$$\Sigma^1_1\text{-}\mathsf{AC}_0 \geq \mathsf{IRT} > \mathsf{ACA}_0 \equiv \text{nonuniform-}\mathsf{WIRT} \geq \text{uniform-}\mathsf{WIRT} > \mathsf{RCA}_0.$$

APPENDIX A

LATTICE THEORY

The various pieces of the theory of lattices required for our applications are collected here. It is standard to denote meet and join by $\wedge$ and $\vee$ respectively, however, so as to not conflict with conjunction and disjunction in formal languages, we use $\sqcap$ and $\sqcup$, respectively. Also, to keep the notation for an order on a lattice and the order on a notion of forcing separate, we use $\sqsubseteq$ for the order relation on lattices.

A.1 Preliminaries

A **partially ordered set** (or **poset**) $\mathscr{P} = \langle P, \sqsubseteq_{\mathscr{P}} \rangle$ is a pair where P is a set and $\sqsubseteq_{\mathscr{P}}$ is a binary relation on P that is

- reflexive: $\forall p \in P(p \sqsubseteq_{\mathscr{P}} p)$;
- transitive: $\forall p, q, r \in P((p \sqsubseteq_{\mathscr{P}} q \wedge q \sqsubseteq_{\mathscr{P}} r) \rightarrow p \sqsubseteq_{\mathscr{P}} r)$;
- and antisymmetric: $\forall p, q \in P((p \sqsubseteq_{\mathscr{P}} q \wedge q \sqsubseteq_{\mathscr{P}} p) \rightarrow p = q)$.

A **preorder** is a pair $\langle P, \sqsubseteq_{\mathscr{P}} \rangle$ satisfying everything in the definition of a poset except it need not be antisymmetric. Given a preorder $\mathscr{P} = \langle P, \sqsubseteq_{\mathscr{P}} \rangle$ one can define the equivalence relation $\equiv_{\mathscr{P}}$ on P given by

$$p \equiv_{\mathscr{P}} q \iff p \sqsubseteq_{\mathscr{P}} q \text{ and } q \sqsubseteq_{\mathscr{P}} p.$$

This induces a quotient structure $\mathscr{Q} = \langle Q, \sqsubseteq_{\mathscr{Q}} \rangle$ where $Q = P/\equiv_{\mathscr{P}}$ and $\sqsubseteq_{\mathscr{Q}}$ is a binary relation on Q given by

$$[p] \sqsubseteq_{\mathscr{Q}} [q] \iff p \sqsubseteq_{\mathscr{P}} q.$$

This relation is well-defined on Q and endows it with the structure of a poset.

If S is a subset of a poset $\mathcal{P}$, then $p \in \mathcal{P}$ is an **upper bound** of S if $q \sqsubseteq_{\mathcal{P}} p$ for each $q \in S$. Such a p is the **least upper bound** (also called the **join** or **supremum**) of S if p is an upper bound of S, and p is least among the upper bounds of S, i.e., if q is an upper bound of S, then $p \sqsubseteq_{\mathcal{P}} q$. The dual notion of **greatest lower bound** (or **meet** or **infimum**) is given by reversing every inequality. Observe that if a meet or join exists for a set S, then it is unique.

A **lattice** is a poset such that every non-empty finite subset has a meet and a join. Frequently, we present a lattice as a structure $\mathscr{L} = \langle L, \sqsubseteq_{\mathscr{L}}, \sqcap_{\mathscr{L}}, \sqcup_{\mathscr{L}} \rangle$ such that $\langle L, \sqsubseteq_{\mathscr{L}} \rangle$ is the underlying poset, and $\sqcap_{\mathscr{L}}$ and $\sqcup_{\mathscr{L}}$ are binary functions from $L \times L$ to L that take a pair of elements and returns the their meet and join, respectively. A lattice is **bounded** if there are $\bot_{\mathscr{L}}, \top_{\mathscr{L}} \in \mathscr{L}$ that are, respectively, the least and greatest elements of $\mathscr{L}$. In a bounded lattice every finite subset of $\mathscr{L}$ has a meet and a join. In the body of this work, we will assume that all lattices are bounded.

An **upper-semilattice** (or **USL)** is a poset such that every non-empty finite subset has a join. Similarly to lattices, we present these as structures $\mathscr{U} = \langle U, \sqsubseteq_{\mathscr{U}}, \sqcup_{\mathscr{U}} \rangle$ where $\sqcup_{\mathscr{U}}$ is the binary join operator. We will additionally assume that USLs have a least element $\bot_{\mathscr{U}}$. We will also consider USLs with a greatest element $\top_{\mathcal{U}}$; we call such a structure an **USL with** $\top$ (**USL**$^{\top}$ for short). There is a dual notion of lower-semilattices given by replacing join with meet, and least elements with greatest element. Observe that a finite semilattice is a lattice.

An **atom** in a bounded lattice (or USL) is an element $x \neq \bot$ such that if $\bot \sqsubseteq y \sqsubseteq x$, then $y = \bot$ or $y = x$. A **coatom** is an element $x \neq \top$ such that if

$x \sqsubseteq y \sqsubseteq \top$, then $y = \top$ or $y = x$.

A.2 Extension theorems

Given a lattice $\mathscr{L}$ (or a semilattice) a **sublattice** (or subsemilattice) is a subset of $\mathscr{L}$ that is closed under the meet and join operation from $\mathscr{L}$ (or in the semilattice case, closed under whichever operation is defined). In particular, a sublattice is not just a subset of $\mathscr{L}$ that is also a lattice; the meets and joins evaluated within the subset must be the same as meets and joins as evaluated within $\mathscr{L}$. Model theoretically, this notion of sublattice coincides with the notion of substructure of $\mathscr{L}$ in the language $\{\sqsubseteq, \sqcap, \sqcup\}$. If our lattice is bounded, then we also required that the least and greatest element are preserved when taking substructures (and similarly with a semilattice and the corresponding distinguished element). A subset of a poset is an **initial segment** if it is closed downwards in the order relation.

For a USL $\mathscr{U}$ and a set X disjoint from $\mathscr{U}$ the **free extension of $\mathscr{U}$ by** X is the USL denoted $\mathscr{U}[X]$ with underlying set $U \times X^{<\omega}$ (i.e., elements of $\mathscr{U}[X]$ are pairs with the first coordinate an element of $\mathscr{U}$ and the second a finite subset of X), order relation defined by

$$\langle u_1, X_1 \rangle \sqsubseteq_{\mathscr{U}[X]} \langle u_2, X_2 \rangle \iff u_1 \sqsubseteq_{\mathscr{U}} u_2 \text{ and } X_1 \subseteq X_2,$$

and join defined by

$$\langle u_1, X_1 \rangle \sqcup_{\mathscr{U}[X]} \langle u_2, X_2 \rangle = \left\langle u_1 \sqcup_{\mathscr{U}[X]} u_2, X_1 \cup X_2 \right\rangle.$$

This structure is merely the poset product of $\mathscr{U}$ with the poset of finite subsets of X. As both of these are USLs with least element the product is also a USL with

least element. Observe that there is a natural embedding of $\mathscr{U}$ into $\mathscr{U}[X]$ given by $u \mapsto (u, \emptyset)$ which realizes $\mathscr{U}$ as an initial segment of $\mathscr{U}[X]$.

For a USL$^\top$ $\mathscr{U}$ there is a related notion of **$\top$-preserving free extension of $\mathscr{U}$ by X** denoted $\mathscr{U}[X]^\top$. It has underlying set

$$(\mathscr{U} \setminus \top) \times X^{<\omega} \cup \{\top\},$$

i.e., an element is either $\top$ or it is a pair with first coordinate an element of u that is not $\top$ and the second coordinate a finite subset of X. The order relation is given by declaring $\top$ to be greater than everything and otherwise using the relation for the free extension of $\mathscr{U}$ by X given above. The join is also defined as above, except when one of the arguments is $\top$, in which case the value of the join is $\top$. There is a natural embedding of $\mathscr{U}$ into $\mathscr{U}[X]^\top$ given by $\top \mapsto \top$ and any $u \neq \top$ is mapped to $(u, \emptyset)$. This realizes $\mathscr{U}$ as an **almost initial segment** of $\mathscr{U}[X]^\top$ (i.e. as a subset consisting of an initial segment of $\mathscr{U}[X]^\top$ along with$\top$).

For a USL (or USL$^\top$) $\mathscr{U}$ and a USL (or USL$^\top$) $\mathscr{V}$ extending $\mathscr{U}$ we say $\mathscr{V}$ is a **simple extension** of $\mathscr{U}$ if it is generated over $\mathscr{U}$ by at most one new element. If $\mathscr{U}$ is a USL, then an extension $\mathscr{V}$ is an **end extension** of $\mathscr{U}$ if no new element in $\mathscr{V}$ is below any element in $\mathscr{U}$ (i.e., if for every $v \in \mathscr{V} \setminus \mathscr{U}$ and $u \in \mathscr{U}$ we have $v \not\sqsubseteq_{\mathscr{V}} u$). If $\mathscr{U}$ is a USL$^\top$ and $\mathscr{V}$ is a USL$^\top$ extension of $\mathscr{V}$, then $\mathscr{V}$ is an **almost end extension** of $\mathscr{U}$ if the only old element of $\mathscr{U}$ above any new element of $\mathscr{V}$ is $\top$ (i.e. if $u \in \mathscr{U}$ and $v \in \mathscr{V} \setminus \mathscr{U}$ satisfies $v \leq_{\mathscr{V}} u$, then $u = \top$). Observe that a free extension of a USL is an end extension, and a $\top$ preserving free extension of a USL$^\top$ is an almost end extension.

In the body of this work we prove various results about extending USL and USL$^\top$ embeddings into degree structures. Here, we collect the algebraic facts that reduce the general problem to some special cases.

Theorem A.1 (Jockusch & Slaman [10]). *If $\mathscr{U}$ is countable USL and $\mathscr{V}$ is a countable end extension of $\mathscr{U}$, then $\mathscr{V}$ is a subUSL of a simple end extension of a free extension of $\mathscr{U}$. Furthermore, if $\mathscr{U}$ and $\mathscr{V}$ are finite, then the intermediate USLs can be chosen to be finite.*

Theorem A.2 (Barnes [3]). *If $\mathscr{U}$ is countable a USL$^\top$ and $\mathscr{V}$ is a countable almost end extension of $\mathscr{U}$, then $\mathscr{V}$ is a subUSL$^\top$ of a simple almost end extension of a $\top$-preserving free extension of $\mathscr{U}$. Furthermore if $\mathscr{U}$ and $\mathscr{V}$ are finite, then the intermediate structures can be chosen to be finite.*

Proof of Theorem A.1. Let $\mathscr{U}$ be a countable USL and $\mathscr{V}$ a countable end extension of $\mathscr{U}$. Pick a set X of new elements of the same cardinality as $\mathscr{V} \setminus \mathscr{U}$ and a bijection g from $\mathscr{V} \setminus \mathscr{U}$ and X. Let $\mathscr{U}_1 = \mathscr{U}[X]$ be the free extension of $\mathscr{U}$ by X. We define a map h from $\mathscr{U}_1$ into $\mathscr{V}$ by

$$h(u, X_1) = u \sqcup_{\mathscr{V}} \bigsqcup_{\mathscr{V}} \{v \in \mathscr{V} \setminus \mathscr{U} : g(v) \in X_1\}.$$

Observe that this is well-defined as the set we take the join over is finite, and, furthermore, observe that h is a USL homomorphism (i.e. h preserves $\sqsubseteq, \sqcup$, and $\bot$).

Let z be a new object, and define $\mathscr{U}_2' = \mathscr{U}_1[\{z\}]$ to be the free extension of $\mathscr{U}_1$ by z; we define $\mathscr{U}_2$ as a quotient of $\mathscr{U}_2'$. We define an equivalence relation $\equiv$ on $\mathscr{U}_2'$ such that $x_1, x_2 \in \mathscr{U}_2'$ satisfy $x_1 \equiv x_2$ iff $x_1 = x_2$ or if there exist $y_1, y_2 \in \mathscr{U}_1$ such that $(y_1, z) = x_1, (y_2, z) = x_2$, and $h(y_1) = h(y_2)$. It can be easily seen that $\equiv$ is an equivalence relation. Additionally, $\equiv$ is a congruence relation for the $\sqcup_{\mathscr{U}_2'}$ structure of $\mathscr{U}_2'$, i.e.,

$$\text{if } x_1 \equiv x_1' \text{ and } x_2 \equiv x_2', \text{ then } x_1 \sqcup_{\mathscr{U}_2'} x_2 \equiv x_1' \sqcup_{\mathscr{U}_2'} x_2'.$$

This can be seen using the fact that h is a homomorphism. Consequently, we can use the join structure of $\mathscr{U}_2'$ to induce a binary function $\sqcup_{\mathscr{U}_2}$ on $\mathscr{U}_2 := \mathscr{U}_2'/\equiv$. We then define $\sqsubseteq_{\mathscr{U}_2}$ and $\bot_{\mathscr{U}_2}$ by

$$[x_1] \sqsubseteq_{\mathscr{U}_2} [x_2] \iff [x_1 \sqcup_{\mathscr{U}_2} x_2] = [x_2] \text{ and } \bot_{\mathscr{U}_2} = [\bot_{\mathscr{U}_2'}].$$

It can be shown that $\sqsubseteq_{\mathscr{U}_2}$ is partial order on $\mathscr{U}_2$ and that $\bot_{\mathscr{U}_2}$ and $\sqcup_{\mathscr{U}_2}$ are the least element and join function for this partial order, i.e., that $\mathscr{U}_2$ is a USL.

We want to show that $\mathscr{U}_2$ is a simple end extension of $\mathscr{U}_1$ and that $\mathscr{V}$ is a subUSL of $\mathscr{U}_2$, or, more precisely, we want to show that there is an embedding of $\mathscr{U}_1$ into $\mathscr{U}_2$ such that the image is an initial segment, and that there is an embedding of $\mathscr{V}$ into $\mathscr{U}_2$ that extends the natural embedding of $\mathscr{U}$ given by $u \mapsto [((u, \emptyset), \emptyset)]$.

The embedding from $\mathscr{U}_1$ into $\mathscr{U}_2$ is given by $y \mapsto [(y, \emptyset)]$. Firstly, this map is injective because there are no nontrivial $\equiv$ congruences among elements of $\mathscr{U}_2'$ of the form $(y, \emptyset)$. Secondly, $\mathscr{U}_2$ is simple over the image of $\mathscr{U}_1$ as it is generated by $[(\bot_{\mathscr{U}_1}, z)]$. To show the image is an initial segment, suppose $[(y_1, Z)] \sqsubseteq_{\mathscr{U}_2} [(y_2, \emptyset)]$ for some $y_1, y_2 \in \mathscr{U}_1$ and $Z \subseteq \{z\}$. We must show that $Z = \emptyset$. By definition of $\sqsubseteq_{\mathscr{U}_2}$ we know $(y_1, Z) \sqcup_{\mathscr{U}_2'} (y_2, \emptyset) \equiv (y_2, \emptyset)$. As there are no nontrivial $\equiv$ congruences involving an element of the form $(y, \emptyset)$ it follows that $Z = \emptyset$ and $y_1 \sqcup_{\mathscr{U}_1} y_2 = y_2$, i.e., that $[(y_1, Z)] = [(y_1, \emptyset)]$ is in the image of the embedding and so in $\mathscr{U}_1$.

Our embedding of $\mathscr{V}$ into $\mathscr{U}_2$ is defined by

$$f(v) = \begin{cases} [((v, \emptyset), \emptyset)] & \text{if } v \in \mathscr{U}; \\ [((\bot_{\mathscr{U}}, g(v)), z)] & \text{if } v \in \mathscr{V} \setminus \mathscr{U}. \end{cases}$$

Clearly f is well-defined and extends the natural embedding of $\mathscr{U}$ to $\mathscr{U}_2$ via $\mathscr{U}_1$. Firstly we show that f is injective. If $v_1, v_2 \in \mathscr{V}$ and $f(v_1) = f(v_2)$, then, as their are no nontrivial $\equiv$ congruences involving an element of $\mathscr{U}_2'$ of the form $(y, \emptyset)$,

either both $v_1, v_2 \in \mathscr{U}$ or both $v_1, v_2 \in \mathscr{V} \setminus \mathscr{U}$. In the first case, as f extends the embedding of $\mathscr{U}$ into $\mathscr{U}_2$, then $v_1 = v_2$. In the second case, by the definition of $\equiv$ it follows that $h(\bot_{\mathscr{U}}, g(v_1)) = h(\bot_{\mathscr{U}}, g(v_2))$, and so

$$\begin{aligned} v_1 &= \bot_{\mathscr{U}} \sqcup_{\mathscr{V}} \bigsqcup_{\mathscr{V}} \{v \in \mathscr{V} \setminus \mathscr{U} : g(v) \in \{g(v_1)\}\} \\ &= h(\bot_{\mathscr{U}}, g(v_1)) \\ &= h(\bot_{\mathscr{U}}, g(v_2)) \\ &= \bot_{\mathscr{U}} \sqcup_{\mathscr{V}} \bigsqcup_{\mathscr{V}} \{v \in \mathscr{V} \setminus \mathscr{U} : g(v) \in \{g(v_2)\}\} \\ &= v_2. \end{aligned}$$

Consequently, f is injective. It is yet to be shown that f is a USL embedding; it suffices to check that f preserves least element and join. As $\bot_{\mathscr{V}} = \bot_{\mathscr{U}} \in \mathscr{U}$ it follows that

$$f(\bot_{\mathscr{V}}) = [((\bot_{\mathscr{U}}, \emptyset), \emptyset)] = [(\bot_{\mathscr{U}_1}, \emptyset)] = [\bot_{\mathscr{U}_2'}] = \bot_{\mathscr{U}_2}.$$

For preservation of join there are three cases: both elements are in $\mathscr{U}$, one is in $\mathscr{U}$ and one in $\mathscr{V} \setminus \mathscr{U}$, and both elements are in $\mathscr{V} \setminus \mathscr{U}$. In the first case it follows from the fact that f extends the embedding from $\mathscr{U}$ into $\mathscr{U}_2$. In the second case we have $u \in \mathscr{U}$ and $v \in \mathscr{V} \setminus \mathscr{U}$. As $\mathscr{V}$ is an end extension of $\mathscr{U}$, then $u \sqcup_{\mathscr{V}} v \in \mathscr{V}$ and so

$$\begin{aligned} f(u) &= [((u, \emptyset), \emptyset)]; \\ f(v) &= [((\bot_{\mathscr{U}}, g(v)), z)]; \\ f(u) \sqcup_{\mathscr{U}_2} f(v) &= [((u, \emptyset), \emptyset) \sqcup_{\mathscr{U}_2'} ((\bot_{\mathscr{U}}, g(v)), z)] = [((u, g(v)), z)]; \text{ and} \\ f(u \sqcup_{\mathscr{V}} v) &= [((\bot_{\mathscr{U}}, g(u \sqcup_{\mathscr{V}} v)), z)]. \end{aligned}$$

It suffices to show that $(u, g(v)) \equiv (\bot_{\mathscr{U}}, g(u \sqcup_{\mathscr{V}} v))$, i.e., that $h(u, g(v)) = h(\bot_{\mathscr{U}}, g(u \sqcup_{\mathscr{V}} v))$. To see this, recall that $h(g(v')) = v'$ for any $v' \in \mathscr{V} \setminus \mathscr{U}$

was shown above. Applying this to $v' = v$ and $v' = u \sqcup_{\mathscr{V}} v$ gives the desired equality.

Finally if we have $v_1, v_2 \in \mathscr{V} \setminus \mathscr{U}$, then it follows $v_1 \sqcup_{\mathscr{V}} v_2 \in \mathscr{V} \setminus \mathscr{U}$. Following the same steps as above, it suffices to show that $h(g(v_1) \sqcup_{\mathscr{U}_1} g(v_2)) = h(g(v_1 \sqcup_{\mathscr{V}} v_2))$. As h is a USL homomorphism this equality follows.

Thus, we have $\mathscr{U}_2$ a simple end extension of (an embedding of) $\mathscr{U}_1$ that is a free extension of (and embedding of) $\mathscr{U}$ with all structures countable. Furthermore, $\mathscr{V}$ embeds into $\mathscr{U}_2$ in a manner that extends the embedding of $\mathscr{U}$. Finally, observe that if $\mathscr{V}$ and $\mathscr{U}$ are finite, then $\mathscr{U}_1$ and $\mathscr{U}_2'$ are finite, and so, consequently is $\mathscr{U}_2$. □

Outline of proof for Theorem A.2. The proof is very close to that of A.1. You start by taking $\mathscr{U}_1$ to be the $\top$-preserving free extension of $\mathscr{U}$ by $\mathscr{V} \setminus \mathscr{U}$ many new generators. Let g be a bijection between $\mathscr{V} \setminus \mathscr{U}$ and the generating set X. Define $h : \mathscr{U}_1 \to \mathscr{V}$ by

$$h(x) = \begin{cases} u \sqcup_{\mathscr{V}} \bigsqcup_{\mathscr{V}} \{v \in \mathscr{V} : g(v) \in X_1\} & \text{if } x = (u, X_1); \\ \top & \text{if } x = \top. \end{cases}$$

The map h is a USL$^\top$ homomorphism, i.e., h preserves $\sqsubseteq, \sqcup, \bot$, and $\top$.

Now let z be a new object and define $\mathscr{U}_2'$ to be the $\top$-preserving extension of $\mathscr{U}_1$ by $\{z\}$. We define an equivalence relation $\equiv$ on $\mathscr{U}_2$' by $y_1 \equiv y_2$ iff $y_1 = y_2$, or there exist $x_1, x_2 \in \mathscr{U}_1$ such that $y_1 = (x_1, z), y_2 = (x_2, z)$ and $h(x_1) = h(x_2)$, or $y_1 = (x_1, z)$, $y_2 = \top$, and $h(x_1) = \top$, or $y_1 = \top$, $y_2 = (x_2, \top)$, and $h(x_2) = \top$. One checks that $\equiv$ is an equivalence relation, and there is no non-trivial equivalences between elements of the form $(x, \emptyset)$ and any other element.

From here the proof proceeds as above: $\equiv$ is a congruence relation for $\sqcup_{\mathscr{U}_2'}$ and so this descends to the quotient, which we call $\mathscr{U}_2$. You then define $\bot, \top$, and $\sqsubseteq$ on the quotient as before and observe $\mathscr{U}_2$ is a USL$^\top$. You then need to check that the natural map of $\mathscr{U}_1$ into $\mathscr{U}_2$ is a USL$^\top$ embedding, and that the image is an almost initial segment. Finally, you need to check that this map can be extended to a USL$^\top$ embedding of $\mathscr{V}$. The definition of this map is

$$f(v) = \begin{cases} [\top] & \text{if } v = \top; [((v, \emptyset), \emptyset)] \text{if } v \in \mathscr{U} \setminus \{\top\}; [(g(v), z)] \text{if } v \in \mathscr{V} \setminus \mathscr{U}. \end{cases}$$

This shows that, indeed, $\mathscr{V}$ is a subUSL$^\top$ of a simple almost end extension ($\mathscr{U}_2$) of a $\top$-preserving free extension ($\mathscr{U}_1$) of $\mathscr{U}$. Furthermore, if $\mathscr{V}$ is finite, then $\mathscr{U}_1, \mathscr{U}_2'$ and $\mathscr{U}_2$ are all finite. □

Theorem A.3 (Jockusch & Slaman [10])**.** *Suppose $\mathscr{U}$ is a USL that satisfies*

$$\left[\bigwedge_{j=1}^{m} d_i \not\sqsubseteq c_i \,\&\, (a \not\sqsubseteq c_i \text{ or } d_i \not\sqsubseteq c_i \sqcup b)\right] \rightarrow (\exists g)\left[b \sqsubseteq a \sqcup g \,\&\, \bigwedge_{j=1}^{m} d_j \not\sqsubseteq c_j \sqcup g \,\&\, \bigwedge_{k=1}^{n} g \not\sqsubseteq e_k\right]$$

for each $a, b, c_i, d_i, e_j \in \mathscr{U}$ where $j = 1, \ldots, m$ and $k = 1, \ldots, n$. Then every embedding from a finite USL into $\mathscr{U}$ extends to an embedding of any simple end extension of the USL.

This theorem appeared in Jockusch and Slaman [10]. Their proof has some minor errors that are corrected here.

Proof. We introduce some terminology. Call a formula φ in the language of USLs **valid** in a USL $\mathscr{V}$ if its universal closure is true in $\mathscr{V}$. Write $\varphi \rightarrow^* \psi$ if $\varphi \rightarrow \psi$ is valid in every USL and $\psi \rightarrow \varphi$ is valid in $\mathscr{U}$. Call a formula **special** if is a conjunction of disjunctions formulas of the form $t_1 \not\sqsubseteq t_2$ for terms t_1, t_2. A formula

φ is **amenable** if there is a special formula ψ with the same free variables as φ such that $\varphi \to^* \psi$.

Observe that the converse of the implication in the theorem statement is valid in every USL, and so the consequent is amenable. We want to show that every formula of the form

$$(\exists g)\left[\bigwedge_{i=1}^{l} b_i \sqsubseteq a_i \sqcup g \,\&\, \bigwedge_{j=1}^{m} d_j \not\sqsubseteq c_j \sqcup g \,\&\, \bigwedge_{k=1}^{n} g \sqcup f_k \not\sqsubseteq e_k\right] \tag{A.1}$$

is amenable. The proof by induction on l (over all choices of variables). Firstly we prove the base case. Suppose $l = 1$ in (A.1). Let φ be the version of (A.1) with the f_ks deleted; φ is the consequent in the hypothesis of the theorem. The hypothesis implies φ is amenable with corresponding special formula ψ: the antecedent in the theorem statement. To see that $\varphi \to \psi$ is valid in every USL, observe that as $d_i \not\sqsubseteq c_i \sqcup g$, then $d_i \not\sqsubseteq c_i$, and further, as $b \sqsubseteq a \cup g$, then if $a \sqsubseteq c_i$ and $d_i \sqsubseteq c_i \sqcup b$ we have $d_i \sqsubseteq c_i \sqcup b \sqsubseteq c_i \sqcup (a \sqcup g) \sqsubseteq c_i \sqcup g$; a contradiction.

Now, over any USL φ implies (A.1) (because if $g \not\sqsubseteq e$, then $g \sqcup f \not\sqsubseteq g$). Consequently, $\psi \to$ (A.1) is valid in $\mathscr{U}$. For the other direction over any USL, observe that the same proof that $\varphi \to \psi$ is valid in any USL still works because that proof did not use the hypothesis that $g \not\sqsubseteq e_k$ (this is, of course, because ψ does not mention e_k). This concludes the base case.

Now suppose the $l > 1$ and (A.1) holds for every value up to l over all choices of every other variable. Note that (A.1) is equivalent to

$$(\exists g_1, \ldots, g_l)\left[\bigwedge_{i=1}^{l} b_i \sqsubseteq a_i \sqcup g_i \,\&\, \bigwedge_{j=1}^{m} d_j \not\sqsubseteq c_j \sqcup \bigsqcup_{i=1}^{l} g_i \,\&\, \bigwedge_{k=1}^{n} \bigsqcup_{i=1}^{l} g_i \not\sqsubseteq e_k\right] \tag{A.2}$$

over every USL. To see this, observe that if (A.1) holds in a lattice, then take $g_1 = \cdots = g_l = g$, if (A.2) holds take $g = \bigsqcup_{i=1}^{l} g_i$. Simple manipulations of

formulas shows that (A.2) is equivalent to

$$(\exists g_n)[b_n \sqsubseteq a_n \sqcup g_n \,\&\, \varphi], \tag{A.3}$$

where φ is the formula

$$(\exists g_1, \ldots, g_{l-1}) \left[\bigwedge_{i=1}^{l-1} b_i \sqsubseteq a_i \sqcup g_i \,\&\, \bigwedge_{j=1}^{m} d_j \not\sqsubseteq c_j \sqcup \bigsqcup_{i=1}^{l} g_i \,\&\, \bigwedge_{k=1}^{n} \bigsqcup_{i=1}^{l} g_i \not\sqsubseteq e_k \right].$$

Re-collapsing the quantifiers in φ (which is valid over any USL) gives us a formula that is amenable by the induction hypothesis. Call the corresponding special formula ψ. Replacing φ in (A.3) with ψ gives us

$$(\exists g_n)[b_n \sqsubseteq a_n \sqcup g_n \,\&\, \psi]. \tag{A.4}$$

Now (A.3)$\rightarrow^*$(A.4) and ψ is a conjunction of disjunctions of negative formulas $t_1 \not\sqsubseteq t_2$. Re-writing the matrix of (A.4) in disjunctive normal form gives us a formula logically equivalent to (A.4). Observe each disjunct of this formula has at most one positive literal occurring, and as we are in disjunctive normal form, we can distribute the existential quantifier over the disjuncts to obtain a formula that is a disjunction of formulas, all of which have a single existential formula and then a matrix that is a conjunction of literals, with at most one positive literal. Such formulas are amenable as they are exactly the base case of the induction. Therefore, we can replace each conjunct in this formula with the corresponding special formula. Following the chain of implications through we have that (A.1) $\rightarrow^*$ a disjunction of special formulas. Re-writing this disjunction in conjunctive normal form shows that (A.1) is amenable.

Now we have this technical result we can talk about extensions of embeddings. Fix $\mathscr{U}$ as in the hypothesis, a finite USL $\mathscr{V}$ embedding into $\mathscr{U}$ via f and a simple end extension $\mathscr{V}'$ of $\mathscr{V}'$. Let $g \in \mathscr{V}'$ be a generator of $\mathscr{V}'$ over $\mathscr{V}$, and so every

$v' \in \mathscr{V}' \setminus \mathscr{V}$ has an expression as $g \sqcup_{\mathscr{V}'} v$ for some $v \in \mathscr{V}$. Introduce variables for each member of $v \in \mathscr{V}$. Let θ be the conjunction of all formulas of the form

$$b \sqsubseteq a \sqcup v, \quad d \not\sqsubseteq c \sqcup g, \quad g \not\sqsubseteq e, \quad f_1 \sqsubseteq f_2 \sqcup f_3, \text{ and } f_2 \not\sqsubseteq f_2 \sqcup f_3$$

that are true in $\mathscr{V}'$ where the a, c, d, e, and f_i are all variables corresponding to members of $\mathscr{V}$. The formula θ completely determines the diagram of $\mathscr{V}'$ because $b \sqcup g \sqsubseteq a \sqcup g$ is equivalent to $b \sqsubseteq a \sqcup g$ and $b \sqcup g \not\sqsubseteq e$ follows from $g \not\sqsubseteq e$ over any USL. Let φ be the formula $(\exists g)\theta$. By the above φ is amenable; let ψ be a corresponding special formula.

Now, φ is true in $\mathscr{V}'$ (with the natural interpretation of the variables), and so ψ is true is $\mathscr{V}'$. Since ψ is quantifier free and only uses variables naming members of $\mathscr{V}$, ψ is true in $\mathscr{V}$. Using f to interpret the variables of ψ in $\mathscr{U}$ we have that ψ is true in $\mathscr{U}$. Since $\varphi \to^* \psi$, φ is true in $\mathscr{U}$ (still using f to interpret variables). Let g' be a witness to the truth of φ. As φ determines the diagram of $\mathscr{V}'$ completely, then we can extend f mapping g to g', and we know we have an embedding of $\mathscr{V}'$. This completes the proof. □

Theorem A.4 (Barnes [2])**.** *Suppose $\mathscr{U}$ is a USL$^\top$ that satisfies*

$$\left[\bigwedge_{j=1}^{m} d_j \not\sqsubseteq c_j \,\&\, (a \not\sqsubseteq c_j \text{ or } d_j \not\sqsubseteq c_j \sqcup b)\right] \to (\exists g < \top)\left[b \sqsubseteq a \sqcup g \,\&\, \bigwedge_{j=1}^{m} d_j \not\sqsubseteq c_j \sqcup g \,\&\, \bigwedge_{k=1}^{n} g \not\sqsubseteq e_k\right]$$

for each $a, b, c_i, d_i \in \mathscr{U}$ where $j = 1, \ldots, m$ and each $e_j < \top$ for $k = 1, \ldots, n$. Then every embedding from a finite USL$^\top$ into $\mathscr{U}$ extends to an embedding of any simple almost end extension of the USL.

As with Theorem A.2, the proof of Theorem A.4 is similar to that of Theorem A.3. You introduce the notion of special and amenable for USL$^\top$s and prove a

similar quantifier elimination theorem. Once this is established the argument is the same.

APPENDIX B

DECISION PROCEDURES

Here we collect together the details of the decisions procedures used in the text.

For a structure $\mathscr{M}$ in a language $\mathcal{L}$ the **theory of** $\mathscr{M}$ denoted $\mathrm{Th}(\mathscr{M})$ is the set of all first order $\mathcal{L}$-sentences that are true in $\mathscr{M}$. The Σ_1 theory of $\mathscr{M}$ is the set of all Σ_1 sentences (i.e. sentences that start with a block of existential quantifiers and are otherwise quantifier free) that are true in $\mathscr{M}$. The Σ_2 theory of $\mathscr{M}$ is the set of all Σ_2 sentences (i.e. sentences that start with a block of existential quantifiers, then a block of universal quantifiers and are otherwise quantifier free) that are true in $\mathscr{M}$.

Theorem B.1 (Jockusch & Slaman [10])**.** *Let $\mathscr{U}$ be an USL. To solve the decision problem for the Σ_2 theory of $\mathscr{U}$ in the language $\{\bot, \sqsubseteq, \sqcup\}$ it suffices to be able to answer all questions of the form:*

> *If $\mathscr{V}$ is a finite USL and $\mathscr{V}_1, \ldots, \mathscr{V}_n$ are finite USL extensions of $\mathscr{V}$, then for every embedding f of $\mathscr{V}$ into $\mathscr{U}$ does there exist an $i \in \{1, \ldots, n\}$ such that f has an extension to an embedding of $\mathscr{V}_i$?*

Proof. To decide the Σ_2 theory it suffices to decide the Π_2 theory and reverse the answers. Fix a Π_2 sentence in the language $\{\sqsubseteq, \sqcup, \bot\}$

$$\varphi_1 := (\forall \overline{x})(\exists \overline{y})\psi(\overline{x}, \overline{y})$$

where $\overline{x} = \langle x_1, \ldots, x_n \rangle, \overline{y} = \langle y_1, \ldots, y_m \rangle$, and ψ is quantifier free.

Firstly, enumerate all USLs $\mathscr{U}_i$ that are generated by at most n elements (not counting $\top$ and $\bot$). As each finite subset needs a join, there are at most 2^{n+2}

many elements in such a USL, and this enumeration is uniformly recursive in n. Say there are $l = l(n)$ many USLs enumerated. For each $\mathscr{U}_i$, take the conjunction $\xi_i(\overline{x})$ of each atomic fact in the language. This is also recursive, and observe that $\xi_i(\overline{x})$ describes $\mathscr{U}_i$ up to isomorphism.

Observe that φ_1 is true in $\mathscr{U}$ iff the following sentence φ_2 is true in $\mathscr{U}$:

$$\varphi_2 := \bigwedge_{i=1}^{l} (\forall \overline{x})(\exists \overline{y})[\xi_i(\overline{x}) \to \psi(\overline{x}, \overline{y})].$$

As if φ_1 is true, then for each choice of $\overline{x}$ there exists an $\overline{y}$ that make the consequent of each implication true, hence each implication is true, and so too the whole conjunction. On the other hand, if φ_2 is true, then choose some values for $\overline{x}$ in $\mathscr{U}$. These $\overline{x}$ generate a subUSL of $\mathscr{U}$ that is isomorphic to one of the $\mathscr{U}_i$. Then, the $\overline{x}$ satisfy $\xi_i(\overline{x})$ and so, as the ith conjunct of φ_2 is true, then there are choices of $\overline{y}$ such that $\xi_i(\overline{x}) \to \psi(\overline{x}, \overline{y})$ true. Therefore, as we have established the antecedent, then we have also established the consequent.

As the transformation from φ_1 to φ_2 was recursive, it suffices to decide all formulas of the form φ_2, and so, it suffices to decide each conjunct separately. So, fix an i and consider the formula $(\forall \overline{x})(\exists \overline{y})[\xi(\overline{x}) \to \psi(\overline{x}, \overline{y})]$ where $\xi_i = \xi$.

Convert ψ into disjunctive normal form so $\psi \leftrightarrow \bigvee_{j=1}^{t} \psi_j(\overline{x}, \overline{y})$ where each $\psi_j(\overline{x}, \overline{y})$ is a conjunction of atomic formulas of our language. Enumerate all the USLs generated by $\overline{x}$ and $\overline{y}$, and delete any of them that are not consistent with any of the $\psi_j(\overline{x}, \overline{y})$ what is left is the list of USLs generated by $\overline{x}$ and $\overline{y}$ that are consistent with at least one of the ψ_j and so, consequently, with ψ. We can delete any ψ_j that are not satisfied in any such USL, as such, ψ_j are inconsistent over the theory of USLs.

As above, for each USLs remaining take the conjunction of all the atomic

facts and list these formulas $\psi_{j,k}$ where $\psi_{j,k}$ is consistent with ψ_j (there may be repetitions) and k ranges from 1 to $s(j)$; the number of USLs in our list consistent with ψ_j. I claim that φ_2 is true in $\mathscr{U}$ iff

$$\varphi_3 := (\forall \overline{x})(\exists \overline{y}) \left[\xi(\overline{x}) \rightarrow \bigvee_{j=1}^{t} \bigvee_{k=1}^{s(t)} \psi_{j,k}(\overline{x}, \overline{y}) \right]$$

is true in $\mathscr{U}$. To see this, recall that $\xi(\overline{x})$ defines the USL generated by $\overline{x}$ up to isomorphism, as too does each $\psi_{j,k}$ for $\overline{x}$ and $\overline{y}$, and each $\psi_{j,k}$ is consistent with ψ_j, which is a disjunct of ψ. Consequently, if φ_2 is true in $\mathscr{U}$ and $\overline{x} \in \mathscr{U}$, then there exists $\overline{y} \in \mathscr{U}$ witnessing the truth of φ_2. If $\overline{x}$ do not satisfy $\xi(\overline{x})$, then φ_3 is vacuously true of those $\overline{x}$. Otherwise, $\psi(\overline{x}, \overline{y})$ is true, and so one of the $\psi_j(\overline{x}, \overline{y})$ is true, and so which ever of the $\psi_{j,k}$ that represent the USL generated by this particular choice of $\overline{x}$ and $\overline{y}$ is true. Thus, one of the disjuncts in the consequent is true and therefore the implication is true.

On the other hand, if φ_3 is true, $\overline{x} \in \mathscr{U}$, and $\overline{x}$ satisfy ξ, then there is a choice of $\overline{y}, j$ and k such that $\psi_{j,k}(\overline{x}, \overline{y})$ is true. As $\psi_{j,k}$ implies ψ_j, then ψ_j is true, and therefore so to is ψ. This completes the proof of the equivalence. Observe that φ_3 was obtained uniformly recursively from φ_2, so it suffices to decide each formula of the form φ_3.

Now, suppose we can decide the extension of embeddings question for $\mathscr{U}$ as stated in the hypotheses. Given a sentence of the form φ_3, we can recursively recover the unique USL $\mathscr{V}$ that is described by ξ, and for each $\psi_{j,k}$ we can similarly recover the unique USL $\mathscr{V}_{j,k}$ described by ξ. If the answer to the question "does every USL embedding of $\mathscr{V}$ extend to one of the $\mathscr{V}_{j,k}$" is yes, then pick some $\overline{x} \in \mathscr{U}$ satisfying ξ. These $\overline{x}$ pick out an isomorphic copy of $\mathscr{V}$ in $\mathscr{U}$, i.e., they induce a USL embedding. This embedding can be extended to one of the $\mathscr{V}_{j,k}$, and consequently, the image of the extended embedding satisfies $\psi_{j,k}(\overline{x}, \overline{y})$. Therefore

the $\overline{y}$ generating this over the copy of $\mathscr{U}$ witness the truth of φ_3. On the other hand, if the answer to the question is no, then there is an embedding of $\mathscr{V}$ into $\mathscr{U}$ that can't be extended to any of the $\mathscr{V}_{j,k}$. Consequently, the $\overline{x}$ that generate the image of this embedding (and so satisfy ξ) have no corresponding witnesses $\overline{y}$ satisfying any of the $\psi_{j,k}$, as such $\overline{y}$ would allow us to construct an extension of the embedding to the corresponding $\mathscr{V}_{j,k}$. □

Theorem B.2 (Barnes [3])**.** *Let $\mathscr{U}$ be an USL$^\top$. To solve the decision problem for the Σ_2 theory of $\mathscr{U}$ in the language $\{\bot, \top, \sqsubseteq, \sqcup\}$ it suffices to be able to answer all questions of the form:*

> *If $\mathscr{V}$ is a finite USL$^\top$ and $\mathscr{V}_1, \ldots, \mathscr{V}_n$ are finite USL$^\top$ extensions of $\mathscr{V}$, then for every embedding f of $\mathscr{V}$ into $\mathscr{U}$ does there exist an $i \in \{1, \ldots, n\}$ such that f has an extension to an embedding of $\mathscr{V}_i$?*

The proof of this theorem is almost identical to that of Theorem B.1 except, throughout, you consider sentences in the language $\{\sqsubseteq, \sqcup, \top, \bot\}$, and consider generating USL$^\top$s . Nothing else changes.

BIBLIOGRAPHY

[1] Ash, C. J. ; Knight, J.
Computable structures and the hyperarithmetical hierarchy
Studies in Logic and the Foundations of Mathematics
Vol. 144, Elsevier Science, 2000

[2] Barnes, J.
On the decidability of the Σ_2 theories of the arithmetic and hyperarithmetic degrees as uppersemilattices
The Journal of Symbolic Logic
Vol. 82, Issue 4, Cambridge University Press, 2017

[3] Barnes, J.
The Σ_2 theory of $\mathcal{D}_h(\leq_h \mathcal{O})$ as an uppersemilattice with least and greatest element is decidable
preprint
`https://arxiv.org/abs/1704.06347`

[4] Chong, C. T. ; Yu, L.
Recursion theory: computational aspects of definability
De Gruyter series in logic and its applications
De Gruyter, 2015

[5] Diestel, R.
Graph Theory
Graduate texts in mathematics
5th edition, Springer, 2017

[6] Feferman, S.
Some applications of the notions of forcing and generic sets
Fundamenta mathematicae
Vol. 56, No. 3, 1965

[7] Feferman, S.; Spector, C.
Incompleteness along paths in progressions of theories
Journal of symbolic logic
Vol. 27, Cambridge University Press, 1962

[8] Halin, R.
Über die Maximalzahl fremder unendlicher Wege in Graphen
Mathematische Nachirichten
Vol. 30, Wiley-VCH, 1965

[9] Harrington, L.; Shore, R. A.; Slaman, T. A.
Σ^1_1 in every real in a Σ^1_1 class of reals is Σ^1_1
Computability and Complexity
A. Day, M. Fellows, N. Greenberg, B. Khoussainov and A. Melnikov eds., Springer-Verlag, 2017

[10] Jockusch, C. G.; Slaman, T. A.
On the Σ_2-theory of the upper semilattice of Turing degrees
The Journal of Symbolic Logic
Vol. 58, Number 1, Association for Symbolic Logic, 1993

[11] Kjos-Hanssen, B.; Shore, R. A.
Lattice initial segments of the hyperdegrees
The Journal of Symbolic Logic
Vol. 75, Cambridge University Press, 2010

[12] Kleene, S. C.
Arithmetical predicates and function quantifiers
Transactions of the American Mathematical Society
Vol. 91, 1955

[13] Kleene, S. C.; Post, E. L.
The upper semi-lattice of degrees of recursive unsolvability
The annuls of mathematics
Vol. 59, 1954

[14] Kreisel, G.
The axiom of choice and the class of hyperarithmetic functions
Koninklijke Nederlandse Akademie van Wetenschappen
Proceedings, series A, vol. 65, 1962

[15] Lerman, M.
Degrees of unsolvability
Perspectives in Mathematical Logic
Omega Series, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1983

[16] Lerman, M.; Shore, R. A.
Decidability and invariant classes for degree structures
Transactions of the American Mathematical Society
Vol 310, No. 2, AMS, 1988

[17] Odifreddi, P.
Forcing and Reducibilities

The Journal of Symbolic Logic
Vol. 48, Number 2, Association for Symbolic Logic, 1983

[18] Rogers, H.
Theory of recursive functions and effective computability
MIT university press
McGraw-Hill, New York, 1967

[19] Sacks, G. E.
Higher recursion theory
Perspectives in Mathematical Logic
Springer-Verlag, Berlin, 1990

[20] Shore, R. A.
The Turing degrees: an introduction
Forcing, Iterated Ultrapowers, and Turing Degrees, Lecture Notes Series
Institute for Mathematical Sciences, National University of Singapore, 29,
World Scientific Publishing, 2015, 39-121.

[21] Shore, R. A.; Slaman, T. A.
Defining the Turing jump
Mathematical Research Letters
6(6):711-722, 1999.

[22] Simpson, M.
Arithmetic degrees: initial segments, ω-REA operators and the ω-jump
Ph.D. thesis
Cornell University, 1985.

[23] Simpson, S. G.
Subsystems of second order arithmetic
Springer-Verlag, 1998

[24] Soare, R. I.
Turing computability: theory and applications
Springer, 2016

[25] Spector, C.
Recursive well-orderings
The Journal of Symbolic Logic
Vol. 20, Association for Symbolic Logic, 1955

[26] Turing, A. M.
Systems of logic based on ordinals
PhD thesis, Princeton University