MEASURING THE RELATIVE COMPLEXITY OF MATHEMATICAL CONSTRUCTIONS AND THEOREMS

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

Jun Le Goh

August 2019

MEASURING THE RELATIVE COMPLEXITY OF MATHEMATICAL CONSTRUCTIONS AND THEOREMS

Jun Le Goh, Ph.D.

Cornell University 2019

We investigate the relative complexity of mathematical constructions and theorems using the frameworks of computable reducibilities and reverse mathematics.

First, we study the computational content of various theorems with reverse mathematical strength around Arithmetical Transfinite Recursion (ATR_0) from the point of view of computable reducibilities, in particular Weihrauch reducibility. We show that it is equally hard to construct an embedding between two given well-orderings, as it is to construct a Turing jump hierarchy on a given well-ordering. We obtain a similar result for Fraïssé's conjecture restricted to well-orderings.

We then turn our attention to König's duality theorem, which generalizes König's theorem about matchings and covers to infinite bipartite graphs. We show that the problem of constructing a König cover of a given bipartite graph is roughly as hard as the following "two-sided" version of the aforementioned jump hierarchy problem: given a *linear* ordering L, construct either a jump hierarchy on L (which may be a pseudohierarchy), or an infinite L-descending sequence. We also obtain several results relating the above problems with choice on Baire space (choosing a path on a given ill-founded tree) and unique choice on Baire space (given a tree with a unique path, produce said path).

Next, we investigate three known ways to formalize the notion of solving a problem by applying other problems in series: the compositional product, the reduction game, and the step product. We clarify the relationships between them

by giving sufficient conditions for them to be equivalent. We also show that they are not equivalent in general.

Next, we turn our attention to the parallel product. In joint work with Dzhafarov, Hirschfeldt, Patey and Pauly, we investigate the infinite pigeonhole principle for different numbers of colors and how these problems behave under Weihrauch reducibility with respect to parallel products.

Finally, we leave the setting of computable reducibilities for the setting of reverse mathematics. First, we define a Σ^1_1 axiom of finite choice and investigate its relationships with other theorems of hyperarithmetic analysis. For one, we show that it follows from Arithmetic Bolzano-Weierstrass. On the other hand, using an elaboration of Steel's forcing with tagged trees, we show that it does not follow from Δ^1_1 comprehension. Second, in joint work with James Barnes and Richard A. Shore, we analyze a theorem of Halin about disjoint rays in graphs. Our main result shows that Halin's theorem is a theorem of hyperarithmetic analysis, making it only the second "natural" (i.e., not formulated using concepts from logic) theorem with this property.

BIOGRAPHICAL SKETCH

Goh Jun Le (吴均乐) was born and raised in Singapore. He was part of the pioneer batch of students in the National University of Singapore High School of Mathematics and Science. In 2008, he enrolled full-time in the National University of Singapore. Before he could graduate, however, he was conscripted for two years. After receiving a B.Sc. (Hons) in mathematics from the National University of Singapore in 2013, he began graduate study at Cornell University in the United States of America, where he is known as Jun Le Goh.

ACKNOWLEDGEMENTS

I am very grateful to my advisor Richard A. Shore for teaching me how to be a researcher. Through his guidance, many half-baked ideas and "fantasies" were realized. I could not have asked for a better mentor.

I am very grateful to Chi Tat Chong for his mentorship and sage advice. He introduced me to the joys of mathematics. This journey of mine would not have gotten off the ground if not for his patience.

I would like to thank my colleagues in computability for their support, advice, and invitations to conferences and research visits. Special thanks goes to Peter Cholak, Barbara Csima, Damir Dzhafarov, Denis Hirschfeldt, Takayuki Kihara, Julia Knight, Steffen Lempp, Alberto Marcone, Joe Miller, Antonio Montalbán, Ludovic Patey, Arno Pauly, Ted Slaman, Reed Solomon, Mariya Soskova and Linda Brown Westrick.

I thank the National Science Foundation, the Association of Symbolic Logic, the Cornell Graduate School, and the American Mathematical Society for their financial support, which enabled me to meet the aforementioned researchers.

To my friends, both local and overseas: Thank you for inviting me to hang out, accepting me for who I am, and keeping me sane. Special shoutout to the Malaysians and Singaporeans in 117, and the math climbing group, with whom I had countless dinners at Souvlaki House.

Finally, I thank my family and my girlfriend Ai Hui for their unconditional and unwavering support through all of the difficult times.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Acknowledgements . . . iv
Table of Contents . . . v
List of Figures . . . vii

1 Introduction **1**
1.1 Reverse mathematics . . . 3
1.2 Other lenses . . . 8
1.3 Computable reducibilities . . . 10
1.3.1 Representations . . . 11
1.3.2 The Weihrauch lattice of problems . . . 12
1.3.3 Other reducibilities . . . 16
1.4 The arithmetical, analytical and hyperarithmetical hierarchies . . . 17

2 Embeddings between well-orderings and ATR **24**
2.1 Background . . . 24
2.2 An ATR-like problem . . . 26
2.3 Theorems about embeddings between well-orderings . . . 33
2.4 An analog of Chen's theorem . . . 38
2.5 Reducing ATR to WCWO . . . 43
2.6 Reducing ATR to $\mathsf{NDS}_{\mathrm{WO}}$ and $\mathsf{NIAC}_{\mathrm{WO}}$ . . . 46

3 König's duality theorem and two-sided problems **49**
3.1 Two-sided problems . . . 49
3.1.1 ATR_2 and variants thereof . . . 55
3.2 König's duality theorem . . . 61
3.2.1 Reducing ATR_2 to KDT . . . 64
3.2.2 Reducing KDT to ATR_2 . . . 82

4 Different ways of composing multivalued functions **87**
4.1 Formalizing compositions . . . 89
4.1.1 Parallel product . . . 89
4.1.2 Compositional product . . . 90
4.1.3 Reduction games . . . 93
4.1.4 Step product . . . 99
4.2 Composing a multivalued function with itself . . . 102
4.3 Finite compositions of arbitrary multivalued functions . . . 109
4.4 The $\equiv^1_{gW}$-lattice . . . 118

5 Parallel products of the infinite pigeonhole principle **120**
5.1 The product coloring is optimal . . . 120
5.2 How many colors can a product of colorings handle? . . . 122

6 A Σ^1_1 axiom of finite choice **135**
6.1 Theories of hyperarithmetic analysis 135
6.2 Arithmetic Bolzano-Weierstrass implies finite-Σ^1_1-AC_0 139
6.3 Δ^1_1-CA_0 does not imply finite-Σ^1_1-AC_0 141
6.3.1 The model 142
6.3.2 The forcing language 143
6.3.3 The forcing notion 145
6.3.4 The forcing relation 147
6.3.5 Analyzing the forcing relation for ranked formulas 148
6.3.6 Analyzing the forcing relation for Σ-over-$\mathcal{L}_F$ formulas 155
6.3.7 M_∞ satisfies Δ^1_1-comprehension 168

7 Halin's theorem on disjoint rays **176**
7.1 The weak infinite ray theorem 178
7.1.1 Upper bounds 179
7.1.2 Lower bounds via computably enumerable equivalence relations 183
7.2 The infinite ray theorem 189

Bibliography **193**

LIST OF FIGURES

1.1 A Weihrauch reduction from P to Q. 13

5.1 Case 2 in Lemma 5.15, assuming that $b_0 = b_1$. 131
5.2 Case 3 in Lemma 5.15. 132

6.1 Arrows correspond to extension in the forcing. Dotted lines correspond to some notion of retagging, which will be made precise in the proof of Lemma 6.26. 159
6.2 Arrows correspond to extension in the forcing. Dotted lines correspond to retaggings. 167
6.3 p and r lie in G, while q "looks like" it lies in G. 171

7.1 Partial zoo of theories of hyp analysis (assuming $I\Sigma^1_1$) 192

CHAPTER 1

INTRODUCTION

Mathematicians often make statements of the following forms: "theorem A is needed to prove theorem B", or "construction A is not sufficient for proving theorem B", or "proof A of this theorem is more direct than proof B". My research explores the mathematical content of such statements by analyzing the relative complexity of mathematical constructions and theorems.

What could it mean for a theorem or construction to be more "complicated" than another? Certainly a special case of a theorem is no more complicated than the theorem itself. More generally, if there is an "easy" proof of theorem A from theorem B or if one can "easily" construct A using B, then A is no more complicated than B.

What, then, is an "easy" proof or construction? We want to avoid triviality (everything is easy) and intractability (everything is complicated): neither extreme has anything useful to say about the mathematics. A happy balance is struck using *computability*, which captures the notion of being algorithmically solvable (e.g., using a sufficiently powerful programming language with unbounded memory). Let us digress briefly to define some basic notions in computability theory.

First, a (possibly partial) function $f :\subseteq \mathbb{N} \to \mathbb{N}$ is *computable* if there is a Turing machine M that simulates it, i.e., for any $x \in \text{dom}(f)$, M eventually halts on input x and outputs $f(x)$; for any other x, M never halts. In particular, a set of natural numbers A is *computable* if membership in A can be decided by a Turing machine, i.e., the characteristic function of A is computable. This is a robust notion that allows us to discuss computability of sets of objects other than numbers (e.g., finite strings of numbers, rational numbers, Diophantine equations, formulas in a finite language, finitely presented groups) via encodings.

By augmenting Turing machines with oracles, we can define relative computability: we say that A is *B-computable* or *computable in B* if A can be computed by a Turing machine with *oracle access* to B, i.e., the Turing machine computing A is given access to answers to questions of the form "is $n \in B$?" at any step of its computation. This induces the notion of *Turing reduction*, written $A \leq_T B$.

We can define relative computability for *total* functions on $\mathbb{N}$ using relative computability for subsets of $\mathbb{N}$, by encoding a function from $\mathbb{N}$ to $\mathbb{N}$ as a set of pairs and using a standard pairing function.

Finally, we define $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ to be *computable* if there is an oracle Turing machine M such that for any $x \in \text{dom}(F)$, $F(x)$ can be computed using M with oracle access to x. Note that the same M has to work for all $x \in \text{dom}(F)$, so this is stronger than merely asserting that $F(x)$ is computable in x for all $x \in \text{dom}(F)$. This notion of uniformity is fundamental for the present work.

Let us now return to consider the complexity of constructions. We may think of a construction as having an input and an output; for instance compactness takes an open cover as input and outputs any finite subcover. Then we might say that a construction is computable if for any input, we can use it as an oracle to compute some corresponding output. Alternatively, we might demand more uniformity: perhaps we want a single oracle machine which, given any input, computes some corresponding output. (For now we content ourselves with vague generalities.) The study of mathematics which only allows computable constructions is known as computable mathematics.[1]

[1]This should not be confused with constructive mathematics; for example, we always work

With computable mathematics as a base (however we choose to define it), we can measure and compare the complexity of theorems and constructions. My work is conducted in two closely related frameworks for doing so, which are built upon the concepts of *proof* and *reduction/translation* respectively.

Chapters 2, 3, 5 and 4 will be conducted in the framework of computable reducibilities, while chapter 6 will be conducted in the framework of reverse mathematics. In the rest of this chapter, we provide background for these frameworks. We start with reverse mathematics; even though the majority of this thesis is not a direct contribution toward reverse mathematics, it serves to motivate and contextualize much of the present work.

1.1 Reverse mathematics

Reverse mathematics begins with the maxim "When the theorem is proved from the right axioms, the axioms can be proved from the theorem." (Friedman, ICM 1974 [18]) In this case, the axioms would be necessary for proving the theorem! This maxim is justified by the remarkable "Big Five" phenomenon: in the decades since, it was found that many basic theorems in algebra, analysis, combinatorics, topology, etc. are provably *equivalent* to one of five systems of axioms, over the base system of RCA_0 (defined below). Furthermore, these five systems are *linearly ordered* in terms of provability. The standard reference for reverse mathematics is Simpson [42].

The basic setup is as follows. First, we fix a language which is sufficiently expressive for formalizing our theorems of interest. The language of set theory certainly suffices, but in fact the language L_2 of second-order arithmetic (defined

with classical logic rather than intuitionistic logic.

below) is already rich enough to formalize many theorems of interest. This includes most theorems about countable objects, and objects that can be represented by countable objects, such as the real numbers. Most of reverse mathematics has been conducted in L_2.

Definition 1.1. L_2 consists of the usual language of first-order arithmetic, augmented with set variables and quantifiers over them, and a binary predicate symbol $\in$, relating numbers and sets. We also have the equality symbol relating sets, which always satisfies extensionality. An L_2-*structure* is a tuple

$$M = (|M|, \mathcal{S}_M, +_M, \cdot_M, 0_M, 1_M, <_M),$$

where $\mathcal{S}_M$ is a set of subsets of $|M|$, $+_M$, $\cdot_M$, and $<_M$ are binary relations on $|M|$, and 0_M and 1_M are elements of $|M|$.

Formulas of L_2 are interpreted in M in the obvious way. In particular, number quantifiers range over $|M|$ and set quantifiers range over $\mathcal{S}_M$. $|M|$ and $\mathcal{S}_M$ are called the *first-order universe* and *second-order universe* of M respectively. (We often write $\mathbb{N}$ instead of $|M|$, and $X \in M$ instead of $X \in \mathcal{S}_M$.)

Given a structure M, we may expand L_2 to include *parameters* from M, i.e., a constant for each element of $\mathcal{S}_M$. They are treated syntactically as free set variables. Formulas with parameters are interpreted in M in the obvious way.

Next, we fix a base theory in our language, which is too weak to prove our theorems outright (hence avoiding triviality), yet strong enough to prove "basic" facts (hence avoiding intractability). The standard base theory is a possible formalization of computable mathematics. It is named RCA_0, after the Recursive Comprehension Axiom below.

Definition 1.2. Apart from basic axioms asserting that $(\mathbb{N}, +, \cdot, 0, 1, <)$ is a commutative ordered semiring with cancellation, RCA_0 consists of:

- the set induction axiom:

$$\forall X(0 \in X \wedge (n \in X \to n+1 \in X) \to \forall n(n \in X));$$

- the Σ^0_1 induction axiom schema:

$$\varphi(0) \wedge (\varphi(n) \to \varphi(n+1)) \to \forall n \varphi(n),$$

  for any $\varphi(n)$ which is Σ^0_1;

- the Δ^0_1 (recursive) comprehension axiom schema:

$$\forall n(\varphi(n) \leftrightarrow \neg\psi(n)) \to \exists X \forall n(n \in X \leftrightarrow \varphi(n)),$$

  for any $\varphi(n)$ and $\psi(n)$ which are Σ^0_1.

Note that being Δ^0_1 is not a syntactic property, hence the necessity of the antecedent in the Δ^0_1 comprehension schema. Note also that the formulas φ and ψ in the latter two schema are allowed to have set parameters. This allows us to apply comprehension relative to sets in a model. For example, if A and B lie in a model M of RCA_0, then we can apply Δ^0_1 comprehension to show that their *join*

$$A \oplus B = \{2n : n \in A\} \cup \{2n+1 : n \in B\}$$

lies in M as well.

Having fixed a base theory, our next step is to fix a theorem P, and investigate what axioms we need to add to our base theory in order to prove P. There are two directions to this investigation. First we need to find a sufficiently strong system T

(typically consisting of set existence axioms, such as comprehension axioms) such that T (plus our base theory) proves P. After doing so, ideally, we want to obtain a *reversal*, i.e., we want to show that P (plus our base theory) proves T. That shows that the axioms T are both sufficient and necessary in order to prove P.

We have already defined one system from the Big Five, namely RCA_0. Another system from the Big Five is ACA_0, named after the Arithmetical Comprehension Axiom below.

Definition 1.3. The system ACA_0 consists of RCA_0 together with the arithmetical comprehension axiom schema, which consists of

$$\exists X \forall n (n \in X \leftrightarrow \varphi(n)),$$

for any $\varphi(n)$ which is arithmetical.

The following theorems are known to be equivalent to ACA_0:

- every infinite finitely branching tree has an infinite path (König's infinity lemma);
- every bounded sequence in $\mathbb{R}$ has a cluster point (Bolzano-Weierstrass);
- every countable commutative ring has a maximal ideal.

Yet another system in the Big Five is Arithmetical Transfinite Recursion (ATR_0), which lies one step above ACA_0. It is equivalent to the following theorems:

- any two countable well-orderings are comparable;
- any uncountable closed subset of $\mathbb{R}$ has a perfect subset;

– König's duality theorem about countable bipartite graphs (defined in section 3.2).

The next step (in the Big Five) above ATR_0 is the system of Π^1_1 Comprehension (Π^1_1-CA_0), which is equivalent to the Cantor-Bendixson theorem: every closed set in $\mathbb{R}$ is the union of a perfect closed set and a countable set. (Sources for all of the above equivalences can be found in Simpson [42].)

We note that there are several exceptions to the Big Five phenomenon, such as Ramsey's theorem and its consequences. In chapter 6, we study several exceptions which lie strictly between ACA_0 and ATR_0.

We end this section by explicating a connection between proof-theoretic strength and computability-theoretic strength. Earlier, we asserted that RCA_0 is a formalization of computable mathematics. One way to make that precise is to restrict ourselves to *ω-models* of second-order arithmetic, which are L_2-structures whose first-order universe is the standard natural numbers (with $+$, $\cdot$, 0, 1, $<$ interpreted in the standard way). An ω-model is determined entirely by its second-order universe.

It can be shown that the ω-models of RCA_0 are exactly those whose second-order universe is closed under Turing reduction and join $\oplus$. This is essentially because for any set $X \subseteq \mathbb{N}$, the sets which are Turing reducible to X are exactly those which are Δ^0_1-definable with X as a parameter. Hence in the context of ω-models, RCA_0 is essentially equivalent to "computable sets exist".

How about noncomputable sets? For that we need systems stronger than RCA_0. A basic example of a noncomputable set is the halting problem for Turing machines. More generally, for any $A \subseteq \mathbb{N}$, the halting problem for Turing machines with oracle access to A is called the *(Turing) jump* of A, denoted A'. By iterating the jump, we can obtain more and more complicated sets (with respect to Turing reducibility). We say that A is *B-arithmetic*, or that A is *arithmetically reducible* to B, if A is Turing reducible to some finite iterate of the jump applied to B. If A is $\emptyset$-arithmetic, we simply say that A is *arithmetic*.

For example, if T is an infinite finitely branching subtree of $\mathbb{N}^{<\mathbb{N}}$ (i.e., an instance of König's lemma), then T need not have a T-computable path, but it must have a T-arithmetic path (in fact, one that is computable in T''.)

It can be shown that the ω-models of ACA_0 are exactly those which are closed under arithmetic reduction and join. This is essentially because for any set $X \subseteq \mathbb{N}$, the sets which are arithmetically reducible to X are exactly those which are definable by an arithmetical formula with X as a parameter. Hence in the context of ω-models, ACA_0 is essentially equivalent to "arithmetic sets exist" or "finite iterates of the jump exist".

1.2 Other lenses

Reverse mathematics is one of many lenses through which we view the zoo of theorems. From its point of view, an optimal proof is one with the least axiomatic assumptions. But such proofs could be suboptimal in other ways. In fact, many theorems are more directly connected than an implication over RCA_0 would suggest. We wish to make these connections explicit where they exist, and prove that they do not exist otherwise.

For example, we can prove König's lemma using the Bolzano-Weierstrass theorem: given a finitely branching tree $T = \{\sigma_n : n \in \mathbb{N}\}$, we can define a sequence $X = \{x_n : n \in \mathbb{N}\}$ in $[0, 1]$ encoding the nodes of T such that any cluster point x of X can be decoded into an infinite path P on T. (The fact that T is finitely branching ensures that every cluster point of X is in the range of the encoding.)

This is an example of a *reduction* from the problem corresponding to König's lemma to the problem corresponding to the Bolzano-Weierstrass theorem: given an instance T of König's lemma, we defined an instance X of Bolzano-Weierstrass such that for any solution (i.e., cluster point) x of X, we can define a solution (i.e., infinite path) P of T. Furthermore, the maps $T \mapsto X$ and $x \mapsto P$ are continuous, computable even. This means that we can uniformly computably *translate* the problem of finding a solution to König's lemma into the problem of finding a solution to Bolzano-Weierstrass.

Not all proofs in reverse mathematics have such a simple form, however. An example is the common proof of the Bolzano-Weierstrass theorem which proceeds by first extracting a monotone subsequence from the given sequence (using a weak form of Ramsey's theorem), and then applying the monotone convergence theorem. In general, a proof could invoke its premises multiple times, either in parallel or in series. (Notice that the former can be simulated by the latter.) If a proof invokes one premise after another, for example, one might ask if one could invoke them in parallel instead, or if one could weaken either of the premises. If a proof invokes a premise more than once, one might ask if that is necessary.

Analogs of the above questions can be studied in a reducibility framework where one could hope to define reducibility notions or algebraic operations which correspond to invoking theorems in parallel or in series. Depending on the situation, we can easily adjust our notion of reducibility to capture the behavior that we wish to study.

1.3 Computable reducibilities

Among the various reducibilities between problems, we focus on Weihrauch reducibility (also known as uniform reducibility). We will define it later (Definition 1.4). For now an example will suffice, namely, the reduction from König's lemma to the Bolzano-Weierstrass theorem which we described earlier.

The framework of uniform reducibility, as its name might suggest, allows us to study nonuniform case divisions in proofs. A basic example is the following proof of the intermediate value theorem: if the given function f has a rational zero, we are done; otherwise we proceed with bisection (which is a computable procedure under the assumption that f has no rational zero). The above proof can be carried out in RCA_0, yet one cannot uniformly compute whether a given continuous function has a rational zero or not. (Indeed, one cannot even uniformly compute if a given function has the value zero at a given point.) Could we get away with a uniform case division, or no case division at all? This question can be formalized as follows: is there a Weihrauch reduction from the problem corresponding to the intermediate value theorem to the identity problem?

The framework of Weihrauch reducibility also allows us to study computational problems which are not commonly thought of as theorems, such as those in computable analysis. An important class of such problems is the class of choice problems. For example, $\mathsf{C}_{[0,1]}$ is the problem of choosing an element from a given nonempty closed subset of $[0, 1]$ (appropriately represented). Many choice problems are closely connected, or even Weihrauch equivalent, to problems which correspond to theorems that have been studied in reverse mathematics. (We will see some examples in chapters 2 and 3.) This sheds new light on the computational content of those theorems.

In the remainder of this section, we present some background on computable reducibilities. For a comprehensive introduction to Weihrauch reducibility, we refer the reader to Brattka, Gherardi, Pauly [8].

1.3.1 Representations

At the beginning of this chapter, we defined computability for elements of $\mathbb{N}^\mathbb{N}$ and functions from $\mathbb{N}^\mathbb{N}$ to $\mathbb{N}^\mathbb{N}$. Those notions of computability can be transferred to other sets (such as the real numbers) via representations. Let X be a set of cardinality at most that of $\mathbb{N}^\mathbb{N}$. A *representation of* X is a surjective (possibly partial) map $\delta :\subseteq \mathbb{N}^\mathbb{N} \to X$. The pair (X, δ) is called a *represented space*. If $\delta(p) = x$ then we say that p is a *(δ-)name* for x. Every $x \in X$ has at least one δ-name. We say that $x \in X$ is *computable* if it has some δ-name which is computable.

If we have two representations δ and δ' of a set X, we say that δ is *computably reducible* to δ' if there is some computable function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ such that for all $p \in \text{dom}(\delta)$, $\delta(p) = \delta'(F(p))$. We say δ and δ' are *computably equivalent* if they are computably reducible to each other. Computably equivalent representations of X induce the same notion of computability on X.

Typically, the spaces X we work with have a standard representation (or encoding), which we will not specify in detail.

1.3.2 The Weihrauch lattice of problems

We begin by identifying problems, such as that of constructing an embedding between two given well-orderings, with (possibly partial) multivalued functions between represented spaces, denoted $P :\subseteq X \rightrightarrows Y$. A theorem of the form

$$(\forall x \in X)(\Theta(x) \to (\exists y \in Y)\Psi(x, y))$$

corresponds to the multivalued function $P :\subseteq X \rightrightarrows Y$ where $P(x) = \{y \in Y : \Psi(x, y)\}$. Note that logically equivalent statements can correspond to different problems.

The *domain* of a problem, denoted $\mathrm{dom}(P)$, is the set of $x \in X$ such that $P(x)$ is nonempty. Note that $\mathrm{dom}(P)$ could be empty, in which case P is called the empty problem. We do not require $\mathrm{dom}(P)$ or the graph of P to be definable in any sense. An element of $\mathrm{dom}(P)$ is called a *P-instance.* If x is a P-instance, an element of $P(x)$ is called a *P-solution to x.*

A *realizer* of a problem P is a (single-valued, possibly partial) function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ which takes any name for a P-instance to a name for one of its P-solutions. Intuitively, P is reducible to Q if one can transform any realizer for Q into some realizer for P. If such a transformation can be done in a uniformly computable way, then P is said to be Weihrauch reducible to Q:

Definition 1.4. P is *Weihrauch reducible* (or *uniformly reducible*) to Q, written $P \leq_W Q$, if there are computable functions $\Phi, \Psi :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ such that:

- given a name p for a P-instance, $\Phi(p)$ is a name for a Q-instance;
- given a name q for a Q-solution to the Q-instance named by $\Phi(p)$, $\Psi(p \oplus q)$ is a name for a P-solution to the P-instance named by p.

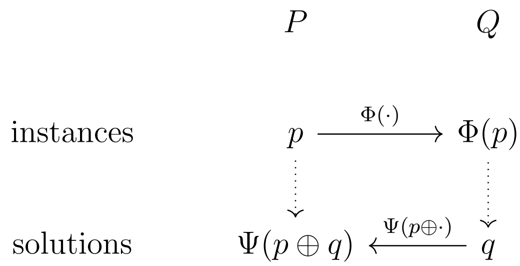

Figure 1.1: A Weihrauch reduction from P to Q.

Figure 1.1 illustrates a Weihrauch reduction from P to Q.

P is *strongly Weihrauch reducible* to Q, written $P \leq_{sW} Q$, if the above holds for some Φ and Ψ where Ψ is not allowed access to p, i.e., $\Psi(q)$ is a name for a P-solution to the given P-instance.

P is *arithmetically Weihrauch reducible* to Q, written $P \leq_W^{\text{arith}} Q$, if the above holds for some arithmetically defined functions Φ and Ψ, or equivalently, some computable functions Φ and Ψ which are allowed access to some fixed finite Turing jump of their inputs.

For any of the above reductions, we say that Φ and Ψ are *forward* and *backward* functionals, respectively, for a reduction from P to Q. We will occasionally use other Greek letters for the forward and backward functionals, such as Γ and Δ.

For readability, we will typically not mention names in our proofs. For example, we will write "given a P-instance" instead of "given a name for a P-instance".

Remark 1.5. Weihrauch reducibility on multivalued functions was first defined by Gherardi and Marcone [20], generalizing earlier work by Brattka and by Weihrauch. (See [7] for historical remarks about Weihrauch reducibility.) Independently, Dorais, Dzhafarov, Hirst, Mileti, and Shafer [15] gave an equivalent definition, and named it uniform reducibility. Our definition follows that in [15].

It is easy to see that Weihrauch reducibility is reflexive and transitive, and hence defines a degree structure on problems. In fact, there are several other natural operations on problems that define corresponding operations on the Weihrauch degrees. For example, the Weihrauch degrees form a distributive lattice (Brattka, Gherardi [6], Pauly [35]) under the following operations:

Definition 1.6. The *join* (or coproduct) of multivalued functions P_0 and P_1, denoted $P_0 \sqcup P_1$, has instances $\bigcup_{i=0,1}\{(i, X) : X \text{ is a } P_i\text{-instance}\}$. For $i = 0, 1$, (i, Y) is a $(P_0 \sqcup P_1)$-solution to (i, X) if Y is a P_i-solution to X.

The *meet* (or sum) of P_0 and P_1, denoted $P_0 \sqcap P_1$, has instances $\{(X_0, X_1) : X_i \text{ is a } P_i\text{-instance}\}$. For $i = 0, 1$, (i, Y) is a $(P_0 \sqcap P_1)$-solution to (X_0, X_1) if Y is a P_i-solution to X_i.

It is easy to see that the join and meet operations lift to the Weihrauch degrees. Next, we have the parallel product, which captures the power of applying problems in parallel:

Definition 1.7 (Brattka, Gherardi [6])**.** The *parallel product* of P and Q, written $P \times Q$, is defined as follows: $\text{dom}(P \times Q) = \text{dom}(P) \times \text{dom}(Q)$ and $(P \times Q)(x, y) = P(x) \times Q(y)$. The (infinite) *parallelization* of P, written $\widehat{P}$, is defined as follows: $\text{dom}(\widehat{P}) = \text{dom}(P)^{\mathbb{N}}$ and $\widehat{P}((x_n)_n) = \{(y_n)_n : y_n \in P(x_n)\}$.

It is easy to see that the parallel product and parallelization operations lift to the Weihrauch degrees. More generally, we can also apply problems in series:

Definition 1.8. The *composition* $\circ$ is defined as follows: for $P :\subseteq X \rightrightarrows Y$ and $Q :\subseteq Y \rightrightarrows Z$, we define $\text{dom}(Q \circ P) = \{x \in X : P(x) \subseteq \text{dom}(Q)\}$ and $(Q \circ P)(x) = \{z \in Z : \exists y \in P(x)(z \in Q(y))\}$.

The composition of problems, however, does not directly induce a corresponding operation on Weihrauch degrees. It is also too restrictive, in the sense that a P-solution is required to be literally a Q-instance. Nevertheless, one can use the composition to define an operation on Weihrauch degrees that more accurately captures the power of applying two problems in series:

Definition 1.9 (Brattka, Gherardi, Marcone [7])**.** The *compositional product* $*$ is defined as follows:

$$Q * P = \sup\{Q_0 \circ P_0 : Q_0 \leq_W Q, P_0 \leq_W P\},$$

where the sup is taken over the Weihrauch degrees.

Brattka and Pauly [9] showed that $Q * P$ always exists.

We end this section by defining some well-studied problems that are helpful for calibrating the problems we are interested in.

Definition 1.10. Define the following problems:

- LPO: given $p \in \mathbb{N}^\mathbb{N}$, output 1 if there is some $k \in \mathbb{N}$ such that $p(k) = 0$, else output 0;
- $\mathsf{C}_\mathbb{N}$: given some $f : \mathbb{N} \to \mathbb{N}$ which is not surjective, output any x not in the range of f;
- $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$: given an ill-founded subtree of $\mathbb{N}^{<\mathbb{N}}$, output any path on it;
- $\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$: given an ill-founded subtree of $\mathbb{N}^{<\mathbb{N}}$ with a unique path, output said path.

For more information about the above problems, we refer the reader to the survey by Brattka, Gherardi, Pauly [8].

1.3.3 Other reducibilities

Apart from arithmetic Weihrauch reducibility (Definition 1.4), we study two other coarsenings of Weihrauch reducibility in this thesis. The first, known as computable reducibility, is a nonuniform version of Weihrauch reducibility:

Definition 1.11 (Dzhafarov [16]). P is *computably reducible* to Q, written $P \leq_c Q$, if given a name p for a P-instance, one can compute a name p' for a Q-instance such that given a name q for a Q-solution to the Q-instance named by p', one can use $p \oplus q$ to compute a name for a P-solution to the P-instance named by p.

For example, even though LPO is not Weihrauch reducible to the identity function, it is computably reducible to the identity because a solution to an LPO-instance is either 0 or 1. The same conclusion holds for $\mathsf{C}_{\mathbb{N}}$.

The second coarsening of Weihrauch reducibility is the notion of generalized Weihrauch reducibility due to Hirschfeldt and Jockusch [24]. Roughly speaking, a generalized Weihrauch reduction from P to Q solves each P-instance using multiple applications of Q in series, in a uniform way. We will only study it in chapter 4, so we define it there instead (Definition 4.9).

Finally, we state an easy proposition which will help us derive corollaries of our results which involve computable reducibility and arithmetic Weihrauch reducibility:

Proposition 1.12. *Suppose* $R \leq_W Q * P$. *If* $Q \leq_c \mathrm{id}$, *then* $R \leq_c P$. *If* $Q \leq_W^{\mathrm{arith}} \mathrm{id}$, *then* $R \leq_W^{\mathrm{arith}} P$.

Observe that the above proposition can be applied with Q being LPO or $\mathsf{C}_{\mathbb{N}}$.

1.4 The arithmetical, analytical and hyperarithmetical hierarchies

We end this chapter by presenting background in recursion theory that will be essential for chapters 2, 3, and 6. For more details on classical recursion theory and hyperarithmetical theory, we refer the reader to Rogers [38] and Sacks [39] respectively.

At the end of section 1.1, we mentioned that the arithmetical subsets of $\mathbb{N}$ are exactly those which are definable by an arithmetical formula. Let us describe the details behind this apparently vacuous statement.

We say that a predicate $R(x, n)$ on $\mathbb{N}^{\mathbb{N}} \times \mathbb{N}$ is *partial recursive* if there is some partial recursive Φ_e such that for all $x \in \mathbb{N}^{\mathbb{N}}$ and $n \in \mathbb{N}$, $R(x, n)$ holds if and only if $\Phi_e^x(n)\downarrow = 0$. We say that $R(x, n)$ is *(total) recursive* if furthermore, for all $x \in \mathbb{N}^{\mathbb{N}}$ and $n \in \mathbb{N}$, $\Phi_e^x(n)\downarrow$.

Using standard pairing functions, we may define what it means for predicates of multiple set and number variables to be partial recursive and total recursive. Now we may define the *arithmetical hierarchy* for predicates, subsets of $\mathbb{N}$, and subsets of $\mathbb{N}^{\mathbb{N}}$: the Σ_1^0 predicates are exactly the partial recursive predicates. A predicate is Π_n^0 if its negation is Σ_n^0. For $n \geq 1$, a predicate P is Σ_{n+1}^0 if there is a Π_n^0 predicate $R(x, m, k)$ such that $P(x, m)$ holds if and only if there is some k such that $R(x, m, k)$ holds. A predicate is Δ_n^0 if it is both Σ_n^0 and Δ_n^0. A predicate is *arithmetical* if it is Σ_n^0 for some n. A subset of $\mathbb{N}$ or $\mathbb{N}^{\mathbb{N}}$ is Σ_n^0 if it is defined by a Σ_n^0 predicate. Likewise for Π_n^0, Δ_n^0, arithmetical, mutatis mutandis.

One can show that the Σ_n^0, Π_n^0, and Δ_n^0 predicates are closed under conjunction, disjunction and bounded quantifiers. The Σ^0_n predicates are closed under existential quantifiers. The Π^0_n predicates are closed under universal quantifiers.

Note that we can relativize all of the above definitions and results by allowing parameters in our predicates. We omit the details.

For subsets of $\mathbb{N}$, Post showed that a set is Σ^0_{n+1} if and only if it is Σ^0_1 relative to the n^{th} jump of the empty set, denoted $\emptyset^{(n)}$. It follows that a subset of $\mathbb{N}$ is Δ^0_{n+1} if and only if it is Δ^0_1 relative to $\emptyset^{(n)}$, or equivalently, $\emptyset'$-computable. Therefore, the subsets of $\mathbb{N}$ which are computable in some finite iterate of the Turing jump are exactly those which are definable by a formula in the language of second-order arithmetic without set quantifiers.

Next, we define the *analytical hierarchy*, which extends the arithmetical hierarchy. A predicate is Σ^1_0 if it is arithmetical. A predicate is Π^1_n if its negation is Σ^1_n. For $n \geq 1$, a predicate $P(x,m)$ is Σ^1_{n+1} if there is a Π^1_n predicate $R(x,y,m)$ such that $P(x,m)$ holds if and only if there is some $y \in \mathbb{N}^{\mathbb{N}}$ such that $R(x,y,m)$. A predicate is Δ^1_n if it is both Σ^1_n and Π^1_n. A predicate is *analytical* if it is Σ^1_n for some n. A subset of $\mathbb{N}$ or $\mathbb{N}^{\mathbb{N}}$ is Σ^1_n if it is defined by a Σ^1_n predicate. Likewise for Π^1_n, Δ^1_n, analytical, mutatis mutandis.

One can show that the Σ^1_n, Π^1_n, and Δ^1_n predicates are closed under conjunction, disjunction and number quantifiers. The Σ^0_n predicates are closed under existential set quantifiers. The Π^0_n predicates are closed under universal set quantifiers.

As before, we can relativize all of the above definitions and results by allowing parameters in our predicates.

In this thesis, we will not go beyond the levels of Σ^1_1 or Π^1_1 in the analytical hierarchy. Of great importance to us is the set W of indices for computable well-orderings. A useful tool for analyzing W (and more) is the Kleene-Brouwer ordering $<_{\mathrm{KB}}$, which linearizes subtrees of $\mathbb{N}^{<\mathbb{N}}$:

Definition 1.13. For any σ and τ in $\mathbb{N}^{<\mathbb{N}}$, $\sigma <_{\mathrm{KB}} \tau$ if σ extends τ, or is to the "left" of τ, i.e., there is $i \in \mathbb{N}$ such that $\sigma \upharpoonright (i-1) = \tau \upharpoonright (i-1)$ and $\sigma(i) < \tau(i)$. If T is a subtree of $\mathbb{N}^{<\mathbb{N}}$, we let $\mathrm{KB}(T)$ denote $<_{\mathrm{KB}} \upharpoonright T$.

Using the Kleene-Brouwer ordering and Kleene's normal form theorem for Π^1_1 predicates, one can show that W is (uniformly) complete among Π^1_1 sets for many-one reducibility (see [39, I.5.4]). A diagonalization argument then shows that W is not Σ^1_1. Analogous results hold for the set of well-orderings (with domain contained in $\mathbb{N}$).

Finally, we introduce the hyperarithmetical hierarchy for subsets of $\mathbb{N}$, which lies between the arithmetical and analytical hierarchy. The idea behind the hyperarithmetical hierarchy is to iterate the Turing jump into the transfinite. First, a definition: the *join* of sets $X_a \subseteq \mathbb{N}$ where $a \in I \subseteq \mathbb{N}$, is the set

$$\bigoplus_{a \in I} X_a = \{\langle a, x \rangle : a \in I, x \in X_a\} \subseteq \mathbb{N},$$

where $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \to \mathbb{N}$ is a standard pairing function. Now for any countable linear ordering L, we say that $(X_a)_{a \in L}$ is a *(Turing) jump hierarchy on* L if for every $b \in L$, X_b is the Turing jump of the join of all X_a such that $a <_L b$. A set A is B-*hyperarithmetic*, or A is *hyperarithmetically reducible* to B, written $A \leq_h B$, if there is a B-computable *well*-ordering L (with first element 0_L) and a jump hierarchy $(X_a)_{a \in L}$ such that $X_{0_L} = B$ and $A \leq_T (X_a)_{a \in L}$. If A is $\emptyset$-hyperarithmetic, we say that it is *hyperarithmetic*. The class of all B-hyperarithmetic sets is denoted $\mathrm{HYP}(B)$. The class $\mathrm{HYP}(\emptyset)$ is simply called HYP.

For example, if P is an isolated path on a subtree T of $\mathbb{N}^{<\mathbb{N}}$, then one can show that P must be T-hyperarithmetic.

Spector showed that the B-hyperarithmetic sets form a hierarchy, stratified by the ordertypes of B-computable well-orderings (see [39, II.4.6]). Essentially, he showed that if L and M are isomorphic B-computable well-orderings, then any jump hierarchies on L and M (with $X_{0_L}, X_{0_M} = B$) are Turing equivalent.

The height of the hyperarithmetical hierarchy, i.e., the least ordinal which is not the ordertype of a computable well-ordering, is denoted ω_1^{CK} (CK stands for Church-Kleene). The least ordinal which is not the ordertype of a B-computable well-ordering is denoted ω_1^B.

The hyperarithmetical hierarchy can be thought of as an effective version of the Borel hierarchy for subsets of $\mathbb{N}^{\mathbb{N}}$. In fact, just as Souslin showed that the Borel hierarchy stratifies the subsets of $\mathbb{N}^{\mathbb{N}}$ which are Δ^1_1-definable with a set parameter, Kleene showed that the B-hyperarithmetical hierarchy stratifies the subsets of $\mathbb{N}$ which are Δ^1_1-definable with B as a parameter (see [39, II.1.4(i) and II.2.5]).

This suggests an analogy between classical recursion theory and hyperarithmetical theory. It is natural to think of enumerating W by a computation of length ω_1^{CK}: at step α, we enumerate all computable well-orderings of length α. Since W is uniformly many-one complete for Π^1_1 sets, we can also think of enumerating any Π^1_1 set by a computation of length up to ω_1^{CK}. If this enumeration halts at some $\alpha < \omega_1^{CK}$, then the Π^1_1 set is in fact hyperarithmetical. This analogy is explored further in the study of metarecursion theory (see [39, V and VI]).

In the remainder of this section, we state several useful results in hyperarithmetical theory.

First, Spector gave a relatively simple proof of Kleene's theorem that HYP = Δ^1_1. The main technical ingredient in Spector's proof is known as boundedness:

Theorem 1.14 (Spector; see [39, I.5.6])**.** *If A is Σ^1_1 and $A \subseteq W$, then there is $\alpha < \omega_1^{CK}$ such that all computable well-orderings with indices in A have length less than α.*

Spector also showed that

Theorem 1.15 (Spector; see [39, II.7.7])**.** *$\omega_1^B > \omega_1^{CK}$ if and only if $W \leq_h B$.*

Another useful result is uniformization for Π^1_1 predicates of numbers, due to Kreisel.

Theorem 1.16 (Kreisel; see [39, II.2.3])**.** *Suppose $P(x, y)$ is a Π^1_1 predicate on $\mathbb{N} \times \mathbb{N}$. Then there is some Π^1_1 predicate $Q(x, y)$ such that (1) for all x, y, $Q(x, y)$ implies $P(x, y)$; (2) for all x for which there is some y such that $P(x, y)$ holds, there is some* unique *z such that $Q(x, z)$ holds. Such Q is said to* uniformize *P.*

Next, we state some basis and "nonbasis" theorems. First, Kleene showed that there is a Σ^1_1 predicate with some solution but no hyperarithmetic solution (see [39, III.1.1]). This is easy once one has shown that the predicate $X \in$ HYP is Π^1_1: consider the Σ^1_1 predicate $X \notin$ HYP.

Another proof of the above fact proceeds via *pseudohierarchies*, which are jump hierarchies on ill-founded computable linear orderings. These were first studied by Harrison [22].

Theorem 1.17 (see [39, III.3.3])**.** *Every pseudohierarchy computes every hyperarithmetical set.*

Let L be an ill-founded computable linear ordering which supports a jump hierarchy. Such linear orderings exist because the class of computable well-orderings is not Σ^1_1, while the class of all computable linear-orderings L which support a jump hierarchy is Σ^1_1. Then the predicate "X is a jump hierarchy on L" is a Σ^1_1 (in fact arithmetic) predicate with solutions, all of which compute every hyperarithmetical set (and hence cannot be hyperarithmetical).

As for basis theorems, Kleene (see [39, III.1.3]) showed that every Σ^1_1 predicate with solutions has a solution $X \leq_T W$. Gandy (see [39, III.1.4]) showed that every Σ^1_1 predicate with solutions has a solution $X <_h W$.

Finally, we formulate a uniform one-to-one correspondence between solutions to arithmetic predicates and Π^0_1 predicates (or equivalently, paths on subtrees of $\mathbb{N}^{<\mathbb{N}}$). This correspondence follows from a proof of Simpson [42, V.5.4], but we give a different proof.

Lemma 1.18. *Given any arithmetic predicate $P(X)$, there is a Π^0_1 predicate $Q(X)$ and a computable bijection F from the solutions of Q to the solutions of P, such that F^{-1} is arithmetic. Furthermore, indices for Q, F, and F^{-1} can be computed uniformly from an index for P.*

Proof. Fix a recursive predicate R and $n \in \mathbb{N}$ such that $P(X)$ holds if and only if $R(X, X^{(n)})$ holds. Start by computing an index for $X^{(n)}$ as a $\Pi^{0,X}_2$ singleton (see [39, II.4.2]). Then define $S(X, Y)$ to be the following Π^0_2 predicate:

$$R(X, Y) \wedge Y = X^{(n)}.$$

Next, define $Q(X, Y, Z)$ as follows:

$$\begin{aligned} & S(X, Y) \\ \wedge\ & Z : \mathbb{N} \to \mathbb{N} \text{ is the minimal Skolem function} \\ & \text{witnessing that } S(X, Y) \text{ holds} \end{aligned}$$

Observe that $Q(X, Y, Z)$ is Π^0_1 as desired. We show that the projection $(X, Y, Z) \mapsto X$ is the desired bijection from solutions of Q to solutions of P.

First, if $Q(X, Y, Z)$ holds, then $P(X)$ holds. Conversely, if $P(X)$ holds, then there is unique (Y, Z) such that $Q(X, Y, Z)$ holds, namely $(X^{(n)}, Z)$ where Z is the minimal Skolem function witnessing that $S(X, Y)$ holds. Furthermore, $(X, X^{(n)}, Z)$ is uniformly computable in $X^{(n+1)}$. $\square$

The above lemma can be generalized to hyperarithmetic predicates, with F^{-1} being hyperarithmetic.

CHAPTER 2

EMBEDDINGS BETWEEN WELL-ORDERINGS AND ATR

In this chapter, we use the framework of computable reducibilities to provide a fine analysis of the computational content of various theorems about embeddings between well-orderings, such as Fraïssé's conjecture for well-orderings and weak comparability of well-orderings. In reverse mathematics, these theorems are known to be equivalent to the system of Arithmetical Transfinite Recursion (ATR_0). Our analysis exposes finer distinctions between these theorems.

First, we define a problem ATR which is analogous to ATR_0 in reverse mathematics (Definition 2.3). Then we show that the problem of computing an embedding between two given well-orderings is as hard as ATR (Theorem 2.30). This answers a question of Marcone [28, Question 5.8]. This also implies that it is no harder to produce an embedding whose range forms an initial segment, than it is to produce an arbitrary embedding.

Note that in this case the situation is the same from the point of view of either Weihrauch reducibility or reverse mathematics. In chapter 3, we will see examples of theorems where the point of view of Weihrauch reducibility is quite different from that of reverse mathematics.

2.1 Background

In this chapter, we will work extensively with the represented spaces of linear orderings and well-orderings, so we describe their representations as follows. If L is a linear ordering or well-ordering whose domain is a subset of $\mathbb{N}$, we represent it as the relation $\{\langle a, b\rangle : a \leq_L b\}$. Then the following operations are computable:

– checking if a given element is in the domain of the ordering;

– adding two given orderings (denoted by $+$);

– adding a given sequence of orderings (denoted by Σ);

– multiplying two given orderings (denoted by $\cdot$);

– restricting a given ordering to a given subset of its domain.

On the other hand, the following operations are not computable:

– checking whether a given element is a successor or limit;

– finding the successor of a given element (if it exists);

– comparing the ordertype of two given well-orderings;

– checking if a given real is a name for a well-ordering.

Next, in many of our proofs, we will use the following version of "effective transfinite recursion" on linear orderings, which easily follows from the recursion theorem. See Sacks [39, I.3.2].

Theorem 2.1. *Let L be an X-computable linear ordering. Suppose $F : \mathbb{N} \to \mathbb{N}$ is total X-computable and for all $e \in \mathbb{N}$ and $b \in L$, if $\Phi_e^X(a)\downarrow$ for all $a <_L b$, then $\Phi_{F(e)}^X(b)\downarrow$. Then there is some e such that $\Phi_e^X \simeq \Phi_{F(e)}^X$. Furthermore:*

– *$\{b : \Phi_e^X(b)\uparrow\}$ is either empty or contains an infinite $<_L$-descending sequence;*

– *Such an index e can be found uniformly in X, an index for F, and an index for L.*

In our applications, X will usually be a sequence of sets $\langle X_a \rangle_a$ indexed by elements of a linear ordering (sometimes L, but not always). We will think of Φ_e^X

as a partial function $f : L \to \mathbb{N}$, and we will think of each $f(b)$ as an index for a computation from some X_a.

2.2 An ATR-like problem

In this section, we formulate a problem which is analogous to ATR_0 in reverse mathematics. Informally, ATR_0 in reverse mathematics asserts that one can iterate the Turing jump along any countable well-ordering starting at any set [42, pg. 38]. We make that precise as follows:

Definition 2.2. Let L be a linear ordering with first element 0_L, and let $A \subseteq \mathbb{N}$. We say that $\langle X_a \rangle_{a \in L}$ is a *jump hierarchy on L which starts with A* if $X_0 = A$ and for all $b >_L 0_L$, $X_b = (\bigoplus_{a <_L b} X_a)'$.

There are several ways to define jump hierarchies. We have chosen the above definition for our convenience. We will show that the Weihrauch degree of the resulting problem is rather robust with regards to which definition we choose. See, for example, Proposition 2.8.

Note that by transfinite recursion and transfinite induction, for any well-ordering L and any set A, there is a unique jump hierarchy on L which starts with A.

Definition 2.3. Define the problem ATR as follows. Instances are pairs (L, A) where L is a well-ordering and $A \subseteq \mathbb{N}$, with unique solution being the jump hierarchy $\langle X_a \rangle_{a \in L}$ which starts with A.

There are significant differences between the problem ATR and the system ATR_0 in reverse mathematics, as expounded in the remark after Theorem 3.2 in Kihara, Marcone, Pauly [28]. For example, in the setting of reverse mathematics, different models may disagree on which linear orderings are well-orderings.

The standard definition of ATR_0 in reverse mathematics [42, Definition V.2.4] involves iterating arbitrary arithmetical operators instead of just the Turing jump. We formulate that statement as a problem and show that it is Weihrauch equivalent to ATR.

Proposition 2.4. ATR *is Weihrauch equivalent to the following problem. Instances are triples* (L, A, Θ) *where* L *is a well-ordering,* $A \subseteq \mathbb{N}$*, and* $\Theta(n, Y, A)$ *is an arithmetical formula whose only free variables are* n*,* Y *and* A*, with unique solution* $\langle Y_a \rangle_{a \in L}$ *such that for all* $b \in L$*,* $Y_b = \{n : \Theta(n, \bigoplus_{a <_L b} Y_a, A)\}$.

Proof. ATR is Weihrauch reducible to the above problem: for the forward reduction, given (L, A), consider (L, A, Θ) where $\Theta(n, Y, A)$ holds if either $Y = \emptyset$ and $n \in A$, or $n \in Y'$. The backward reduction is the identity.

Conversely, given (L, A, Θ), let k be one greater than the number of quantifier alternations in Θ. Apply ATR to $(1 + k \cdot L, L \oplus A)$ to obtain the jump hierarchy $\langle X_\alpha \rangle_{\alpha \in 1+k \cdot L}$.

For the backward reduction, we will use $\langle X_{(a,k-1)} \rangle_{a \in L}$-effective transfinite recursion along L to define a total $\langle X_{(a,k-1)} \rangle_{a \in L}$-recursive function $f : L \to \mathbb{N}$ such that:

- $\Phi_{f(b)}^{X_{(b,k-1)}}$ is total for all $b \in L$;
- if we define $Y_b = \Phi_{f(b)}^{X_{(b,k-1)}}$ for all $b \in L$, then $Y_b = \{n : \Theta(n, \bigoplus_{a <_L b} Y_a, A)\}$.

For each $b \in L$, we define $\Phi_{f(b)}^{X_{(b,k-1)}}$ as follows. First note that $X_{(b,0)}$ uniformly computes $L \oplus A$ (because of the 1 in front of $1 + k \cdot L$), and hence uniformly computes $A \oplus \bigoplus_{a<_L b} X_{(a,k-1)}$. Now $X_{(b,k-1)}$ uniformly computes $X^{(k)}_{(b,0)}$, which uniformly computes $\left(A \oplus \bigoplus_{a<_L b} X_{(a,k-1)}\right)^{(k)}$. Since $\Phi^{X_{(a,k-1)}}_{f(a)}$ is total for all $a <_L b$, that in turn uniformly computes $\left(A \oplus \bigoplus_{a<_L b} Y_a\right)^{(k)}$, where Y_a is defined to be $\{n : \Phi^{X_{(a,k-1)}}_{f(a)}(n)\downarrow= 1\}$. Finally, $\left(A \oplus \bigoplus_{a<_L b} Y_a\right)^{(k)}$ uniformly computes $\{n : \Theta(n, \bigoplus_{a<_L b} Y_a, A)\}$, which defines $\Phi^{X_{(b,k-1)}}_{f(b)}$ as desired.

By transfinite induction along L, f is total. Hence we can compute $Y_b = \Phi^{X_{(b,k-1)}}_{f(b)}$ for all $b \in L$, and output $\langle Y_b \rangle_{b\in L}$. □

When we define reductions from ATR to other problems by effective transfinite recursion, we will often want to perform different actions at the first step, successor steps, and limit steps. If we want said reductions to be uniform, we want to be able to compute which step we are in. This motivates the following definition:

Definition 2.5. A *labeled well-ordering* is a tuple $\mathcal{L} = (L, 0_L, S, p)$ where L is a well-ordering, 0_L is the first element of L, S is the set of all successor elements in L, and $p : S \to L$ is the predecessor function.

We show that when defining Weihrauch reductions from ATR to other problems, we may assume that the given well-ordering has labels:

Proposition 2.6. ATR *is Weihrauch equivalent to the following problem. Instances are pairs* $(\mathcal{L}, A)$ *where* $\mathcal{L} = (L, 0_L, S, p)$ *is a labeled well-ordering and* $A \subseteq \mathbb{N}$*, with unique solution being the jump hierarchy* $\langle X_a \rangle_{a\in L}$ *which starts with* A*.*

Proof. Given (L, A), we can uniformly compute labels for $\omega\cdot(1+L)$. Then apply the above problem to $(\omega\cdot(1+L), L\oplus A)$ to obtain the jump hierarchy $\langle X_{(n,\alpha)} \rangle_{n\in\omega,\alpha\in 1+L}$ which starts with $L \oplus A$.

For the backward reduction, we will use $\langle X_{(0,b)}\rangle_{b\in L}$-effective transfinite recursion along L to define a total $\langle X_{(0,b)}\rangle_{b\in L}$-recursive function $f : L \to \mathbb{N}$ such that $\Phi_{f(b)}^{X_{(0,b)}}$ is total for every $b \in L$ and $\langle \Phi_{f(b)}^{X_{(0,b)}}\rangle_{b\in L}$ is the jump hierarchy on L which starts with A.

First note that every $X_{(0,b)}$ uniformly computes $(L \oplus A)'$, and hence 0_L. This means that it uniformly computes the case division in the following construction.

For the base case, $X_{(0,0_L)}$ uniformly computes $L \oplus A$ and hence A. As for $b >_L 0_L$, $X_{(0,b)}$ uniformly computes L, hence it uniformly computes $(\bigoplus_{a<_L b} X_{(0,a)})'$. Therefore it uniformly computes $(\bigoplus_{a<_L b} \Phi_{f(a)}^{X_{(0,a)}})'$. □

The following closure property will be useful for proving Proposition 2.15. This fact also follows from the combination of work of Pauly ($\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$ is parallelizable [36]) and Kihara, Marcone, Pauly ($\mathsf{ATR} \equiv_W \mathsf{UC}_{\mathbb{N}^\mathbb{N}}$ [28]), but we provide a short direct proof.

Proposition 2.7. ATR *is parallelizable, i.e.,* $\widehat{\mathsf{ATR}} \equiv_W \mathsf{ATR}$.

Proof. It suffices to show that $\widehat{\mathsf{ATR}} \leq_W \mathsf{ATR}$. Instead of $\widehat{\mathsf{ATR}}$, we consider the parallelization of the version of ATR in Proposition 2.6. Given $\langle(\mathcal{L}_i, A_i)\rangle_i$, apply ATR to $(\sum_i L_i, \bigoplus_i L_i \oplus A_i)$ to obtain the jump hierarchy $\langle X_{(i,a)}\rangle_{i\in\omega,a\in L_i}$ which starts with $\bigoplus_i L_i \oplus A_i$.

For each i, we show how to compute the jump hierarchy $\langle X_a\rangle_{a\in L_i}$ which starts with A_i using $(\mathcal{L}_0 \oplus \mathcal{L}_i \oplus \langle X_{(i,a)}\rangle_{a\in L_i})$-effective transfinite recursion along L_i. This is done by defining a total $(\mathcal{L}_0 \oplus \mathcal{L}_i \oplus \langle X_{(i,a)}\rangle_{a\in L_i})$-recursive function $f_i : L_i \to \mathbb{N}$ such that for all $a \in L_i$, $\Phi_{f(a)}^{X_{(i,a)}}$ is total and defines X_a. (The role of $\mathcal{L}_0 \oplus \mathcal{L}_i$ is to provide the values of 0_{L_0} and 0_{L_i} in the following computation.)

For the base case, $X_{(i,0_{L_i})}$ uniformly computes $X_{(0,0_{L_0})} = \bigoplus_i L_i \oplus A_i$, which uniformly computes A_i.

For $b >_{L_i} 0_{L_i}$, $X_{(i,b)}$ uniformly computes $X_{(0,0_{L_0})}$ which uniformly computes L_i, so $X_{(i,b)}$ uniformly computes $(\bigoplus_{a<_{L_i}b} X_{(i,a)})'$. That in turn uniformly computes $(\bigoplus_{a<_{L_i}b} \Phi_{f(a)}^{X_{(i,a)}})' = (\bigoplus_{a<_{L_i}b} X_a)' = X_b$ as desired. □

Henceforth we will primarily work with the following version of ATR:

Proposition 2.8. ATR *is Weihrauch equivalent to the following problem: instances are pairs* $(\mathcal{L}, c)$ *where* $\mathcal{L}$ *is a labeled well-ordering and* $c \in L$*, with unique solution being* Y_c*, where* $\langle Y_a \rangle_{a \in L}$ *is the unique hierarchy such that:*

- $Y_{0_L} = \mathcal{L}$;
- *if* b *is the successor of* a*, then* $Y_b = Y_a'$;
- *if* b *is a limit, then* $Y_b = \bigoplus_{a<_L b} Y_a$.

Proof. Using Proposition 2.4, it is easy to see that the above problem is Weihrauch reducible to ATR.

Conversely, we reduce the version of ATR in Proposition 2.6 to the above problem. Given $(\mathcal{L}, A)$, define

$$M = \omega \cdot (1 + (A, <_{\mathbb{N}}) + L + 1) + 1.$$

Formally, the domain of M is

$$\{(0,n) : n \in \omega\} \cup \{(1,m,n) : m \in A, n \in \omega\}$$
$$\cup\{(2,a,n) : a \in L, n \in \omega\} \cup \{(3,n) : n \in \omega\} \cup \{m_M\}$$

with the ordering described above. It is easy to see that $L \oplus A$ uniformly computes M and labels for it. Let $\mathcal{M}$ denote the tuple of M and its labels.

Apply the given problem to $\mathcal{M}$ and $m_M \in M$ to obtain Y_{m_M}. Note that since m_M is a limit, Y_{m_M} uniformly computes $Y_{(0,0)} = \mathcal{M}$, and hence $\langle Y_c \rangle_{c \in M}$.

For the backward functional, we perform $(\mathcal{L} \oplus \langle Y_c \rangle_{c \in M})$-effective transfinite recursion along L to define a total $(\mathcal{L} \oplus \langle Y_c \rangle_{c \in M})$-recursive function $f : L \to \mathbb{N}$ such that for each $a \in L$, $\Phi_{f(a)}^{Y_{(2,a,1)}}$ is total and defines the a^{th} column X_a of the jump hierarchy on L which starts with A. Note that $\mathcal{L}$ uniformly computes the following case division.

For the base case, first use $Y_{(2,0_L,1)} = Y'_{(2,0_L,0)}$ to compute $Y_{(2,0_L,0)}$. Now $(2, 0_L, 0)$ is a limit, so $Y_{(2,0_L,0)}$ uniformly computes $Y_{(0,0)} = \mathcal{M}$, which uniformly computes A as desired.

For $b >_L 0_L$, since $(2, b, 0)$ is a limit, $Y_{(2,b,0)}$ uniformly computes $Y_{(0,0)} = \mathcal{M}$, which uniformly computes L. Therefore $Y_{(2,b,0)}$ uniformly computes $\bigoplus_{a <_L b} Y_{(2,a,1)}$, and hence $\bigoplus_{a <_L b} \Phi_{f(a)}^{Y_{(2,a,1)}} = \bigoplus_{a <_L b} X_a$. Therefore $Y_{(2,b,1)}$ uniformly computes $X_b = (\bigoplus_{a <_L b} X_a)'$ as desired.

This completes the definition of f, and hence the reduction from the version of ATR in Proposition 2.6 to the given problem. □

Thus far, we have seen that the Weihrauch degree of ATR is fairly robust with respect to the type of jump hierarchy that it outputs (Propositions 2.4, 2.6, 2.8). However, we still require some level of uniformity in the jump hierarchy produced:

Proposition 2.9. *The problem of producing the Turing jump of a given set is not Weihrauch reducible to the following problem: instances are pairs* (L, A) *where* L

is a well-ordering and $A \subseteq \mathbb{N}$, and solutions to L are hierarchies $\langle X_a \rangle_{a \in L}$ where $X_{0_L} = A$ and for all $a <_L b$, $X'_a \leq_T X_b$. Hence ATR *is not Weihrauch reducible to the latter problem either.*

Proof. Towards a contradiction, fix forward and backward Turing functionals Γ and Δ witnessing otherwise. We will show that Γ and Δ could fail to produce $\emptyset'$ from $\emptyset$. First, $\Gamma^\emptyset$ defines some computable (L, A). We claim that there are finite $\langle \sigma_a \rangle_{a \in L}$ and e such that $\sigma_{0_L} \prec A$ and $\Delta^{\emptyset \oplus \langle \sigma_a \rangle_{a \in L}}(e)\downarrow \neq \emptyset'(e)$.

Suppose not. Then for each e, we may compute $\emptyset'(e)$ by searching for $\langle \sigma_a \rangle_{a \in L}$ such that $\sigma_{0_L} \prec A$ and $\Delta^{\emptyset \oplus \langle \sigma_a \rangle_{a \in L}}(e)\downarrow$. Such $\langle \sigma_a \rangle_{a \in L}$ must exist because if $\langle X_a \rangle_{a \in L}$ is a hierarchy on L which starts with A (as defined in the proposition), then $\Delta^{\emptyset \oplus \langle X_a \rangle_{a \in L}}$ is total. This is a contradiction, thereby proving the claim.

Fix any $\langle \sigma_a \rangle_{a \in L}$ which satisfies the claim. It is clear that $\langle \sigma_a \rangle_{a \in L}$ can be extended to a solution $\langle X_a \rangle_{a \in L}$ to (L, A) for the given problem (e.g., by extending using columns of the usual jump hierarchy). But $\Delta^{\emptyset \oplus \langle X_a \rangle_{a \in L}} \neq \emptyset'$, contradiction. □

If we are willing to allow arithmetic Weihrauch reductions, then ATR remains robust:

Proposition 2.10. ATR *is arithmetically Weihrauch reducible (hence arithmetically Weihrauch equivalent) to the problem in Proposition 2.9.*

For the proof, we refer to the reader to the proof of Proposition 3.13 later. (The only difference is that we use transfinite induction along the given well-ordering to show that we always output a jump hierarchy.)

2.3 Theorems about embeddings between well-orderings

There are several theorems about embeddings between well-orderings which lie around ATR_0 in reverse mathematics. Friedman (see [42, notes for Theorem V.6.8, pg. 199]) showed that comparability of well-orderings is equivalent to ATR_0. Friedman and Hirst [19] then showed that weak comparability of well-orderings is also equivalent to ATR_0. We formulate those two theorems about embeddings as problems:

Definition 2.11. Define the following problems:

- CWO: Given a pair of well-orderings, produce an embedding from one of them onto an initial segment of the other.
- WCWO: Given a pair of well-orderings, produce an embedding from one of them into the other.

Marcone proved the analog of Friedman's result for (strong) Weihrauch reducibility:

Theorem 2.12 (see Kihara, Marcone, Pauly [28])**.** $\mathsf{CWO} \equiv_{sW} \mathsf{UC}_{\mathbb{N}^\mathbb{N}} \equiv_{sW} \mathsf{ATR}$.

In Theorem 2.30, we prove the analog of Friedman and Hirst's result for Weihrauch reducibility, i.e., $\mathsf{WCWO} \equiv_W \mathsf{UC}_{\mathbb{N}^\mathbb{N}}$. This answers a question of Marcone [28, Question 5.8].

Another class of examples of theorems about embeddings comes from Fraïssé's conjecture (proved by Laver [29]), which asserts that the set of countable linear orderings is well-quasi-ordered (i.e., any infinite sequence contains a weakly increasing pair) by embeddability. Shore [40] studied the reverse mathematics of various restrictions of Fraïssé's conjecture. We formulate them as problems:

Definition 2.13. Define the following problems:

$\mathsf{WQO}_{\mathrm{LO}}$: Given a sequence $\langle L_i \rangle$ of linear orderings, produce $i < j$ and an embedding from L_i into L_j.

$\mathsf{WQO}_{\mathrm{WO}}$: Given a sequence $\langle L_i \rangle$ of well-orderings, produce $i < j$ and an embedding from L_i into L_j.

$\mathsf{NDS}_{\mathrm{WO}}$: Given a sequence $\langle L_i \rangle$ of well-orderings, and embeddings $\langle F_i \rangle$ from each L_{i+1} into L_i, produce $i < j$ and an embedding from L_i into L_j.

$\mathsf{NIAC}_{\mathrm{WO}}$: Given a sequence $\langle L_i \rangle$ of well-orderings, produce i and j (we may have $i > j$) and an embedding from L_i into L_j.

$\mathsf{NDS}_{\mathrm{LO}}$ and $\mathsf{NIAC}_{\mathrm{LO}}$ can be defined analogously, but we will not study them.

$\mathsf{WQO}_{\mathrm{LO}}$ corresponds to Fraïssé's conjecture. $\mathsf{WQO}_{\mathrm{WO}}$ is the restriction of Fraïssé's conjecture to well-orderings. $\mathsf{NDS}_{\mathrm{WO}}$ asserts that there is no infinite strictly descending sequence of well-orderings. $\mathsf{NIAC}_{\mathrm{WO}}$ asserts that there is no infinite antichain of well-orderings.

The definitions immediately imply that

Proposition 2.14.

$$\mathsf{NDS}_{\mathrm{WO}} \leq_W \mathsf{WQO}_{\mathrm{WO}} \leq_W \mathsf{WQO}_{\mathrm{LO}}$$

$$\mathsf{NIAC}_{\mathrm{WO}} \leq_W \mathsf{WCWO} \leq_W \mathsf{CWO}$$

$$\mathsf{NIAC}_{\mathrm{WO}} \leq_W \mathsf{WQO}_{\mathrm{WO}}$$

It is not hard to show that all of the problems in Proposition 2.14, except for $\mathsf{WQO}_{\mathrm{LO}}$, are Weihrauch reducible to ATR. (We defer our analysis of the strength of $\mathsf{WQO}_{\mathrm{LO}}$ to section 3.1. See Corollary 3.10.)

Proposition 2.15. $\mathsf{CWO} \leq_W \mathsf{ATR}$ *and* $\mathsf{WQO}_{\mathrm{WO}} \leq_W \mathsf{ATR}$.

Proof. Let Q denote the following apparent strengthening of CWO: a Q-instance is a pair of well-orderings (L, M), and a Q-solution consists of both a CWO-solution F to (L, M) and an indication of whether $L < M$, $L \equiv M$, or $L > M$. Clearly $\mathsf{CWO} \leq_W Q$. (Marcone showed that $\mathsf{CWO} \equiv_W \mathsf{ATR}$ (Theorem 2.12), so actually $\mathsf{CWO} \equiv_W Q$.)

We start by showing that $Q \leq_W \mathsf{ATR}$. Given (L, M), define N by adding a first element 0_N and a last element m_N to L. Apply the version of ATR in Proposition 2.4 to obtain a hierarchy $\langle X_a \rangle_{a \in N}$ such that:

- $X_{0_N} = L \oplus M$;
- for all $b >_N 0_N$, $X_b = \left(\bigoplus_{a <_N b} X_a\right)'''$.

For the backward reduction, we start by using $\langle X_a \rangle_{a \in L}$-effective transfinite recursion along L to define a total $\langle X_a \rangle_{a \in L}$-recursive function $f : L \to \mathbb{N}$ such that $\{(a, \Phi^{X_a}_{f(a)}(0)) \in L \times M : \Phi^{X_a}_{f(a)}(0)\downarrow\}$ is an embedding of an initial segment of L into an initial segment of M.

To define f, if we are given any $b \in L$ and $f \upharpoonright \{a : a <_L b\}$, we need to define $f(b)$, specifically $\Phi^{X_b}_{f(b)}(0)$. Use $X_b = (\bigoplus_{a <_L b} X_a)'''$ to compute whether there is an M-least element above $\{\Phi^{X_a}_{f(a)}(0) : a <_L b\}$ (equivalently, whether $M \backslash \{\Phi^{X_a}_{f(a)}(0) : a <_L b\}$ is nonempty). If so, we output said M-least element; otherwise diverge. This completes the definition of $\Phi^{X_b}_{f(b)}(0)$.

Apply the recursion theorem to the definition above to obtain a partial $\langle X_a \rangle_{a \in L}$-recursive function $f : L \to \mathbb{N}$. Now, to complete the definition of the backward reduction we consider the following cases.

Case 1. f is total. Then we output $\{(a, \Phi^{X_a}_{f(a)}(0)) : a \in L\}$, which is an embedding from L onto an initial segment of M.

Case 2. Otherwise, $\{\Phi^{X_a}_{f(a)}(0) : a \in L, \Phi^{X_a}_{f(a)}(0)\downarrow\} = M$. Then we output $\{(\Phi^{X_a}_{f(a)}(0), a) : a \in L, \Phi^{X_a}_{f(a)}(0)\downarrow\}$, which is an embedding from M onto an initial segment of L.

Finally, note that the last column X_{m_N} of $\langle X_a\rangle_{a\in N}$ can compute which case holds and compute the appropriate output for each case. If Case 1 holds but not Case 2, then $L < M$. If Case 2 holds but not Case 1, then $L > M$. If both Case 1 and 2 hold, then $L \equiv M$.

Next, we turn our attention to $\mathsf{WQO}_{\mathrm{WO}}$. Observe that $\mathsf{WQO}_{\mathrm{WO}} \leq_W \widehat{Q}$: given a sequence $\langle L_i\rangle$ of well-orderings, apply Q to each pair (L_i, L_j), $i < j$. Search for the least (i, j) such that Q provides an embedding from L_i into L_j, and output accordingly.

Finally, $\widehat{Q} \leq_W \widehat{\mathsf{ATR}} \equiv_W \mathsf{ATR}$ (Proposition 2.7), so $\mathsf{WQO}_{\mathrm{WO}} \leq_W \mathsf{ATR}$ as desired. □

In the next few sections, we work toward some reversals. Central to a reversal (say, from WCWO to ATR) is the ability to encode information into well-orderings such that we can extract information from an arbitrary embedding between them. Shore [40] showed how to do this if the well-orderings are indecomposable (and constructed appropriately).

Definition 2.16. A well-ordering X is *indecomposable* if it is embeddable in all of its final segments.

Indecomposable well-orderings also played an essential role in Friedman and

Hirst's [19] proof that WCWO implies ATR_0 in reverse mathematics.

We state two useful properties about indecomposable well-orderings. First, it is easy to show by induction that:

Lemma 2.17. *If M is indecomposable and L_i, $i < n$ each embed strictly into M, then $\left(\sum_{i<n} L_i\right) + M \equiv M$.*

Second, the following lemma will be useful for extracting information from embeddings between orderings.

Lemma 2.18. *Let L be a linear ordering and let M be an indecomposable well-ordering which does not embed into L. If F embeds M into a finite sum of L's and M's, then the range of M under F must be cofinal in some copy of M.*

Therefore, if $M \cdot k$ embeds into a finite sum of L's and M's, then there must be at least k many M's in the sum.

Proof. There are three cases regarding the position of the range of M in the sum. Case 1. F maps some final segment of M into some copy of L. Since M is indecomposable, it follows that M embeds into L, contradiction. Case 2. F maps some final segment of M into a bounded segment of some copy of M. Since M is indecomposable, that implies that M maps into a bounded segment of itself. This contradicts well-foundedness of M. Case 3. The remaining case is that the range of M is cofinal in some copy of M, as desired. □

We remark that for our purposes, we do not need to pay attention to the computational content of the previous two lemmas. In addition, unlike in reverse mathematics, we do not need to distinguish between "M does not embed into L" and "L strictly embeds into M".

2.4 An analog of Chen's theorem

In this section, given a labeled well-ordering $\mathcal{L} = (L, 0_L, S, p)$, $\langle Y_a \rangle_{a \in L}$ denotes the unique hierarchy on L, as defined in Proposition 2.8. (This notation persists for the next two sections, which use results from this section.)

We present the technical ingredients needed for our reductions from ATR to theorems about embeddings between well-orderings. The main result is an analog of the following theorem of Chen, which suggests a bridge from computing jump hierarchies to comparing well-orderings. We will not need Chen's theorem so we will not define the notation therein; see Shore [40, Theorem 3.5] for details.

Theorem 2.19 (Chen [11, Corollary 10.2])**.** *Fix $x \in \mathcal{O}$. There is a recursive function $k(a, n)$ such that for all $a <_{\mathcal{O}} x$ and $n \in \mathbb{N}$,*

1. *$k(a, n)$ is an index for a recursive well-ordering $K(a, n)$;*
2. *if $n \in H_a$, then $K(a, n) + 1 \leq \omega^{|x|}$;*
3. *if $n \notin H_a$, then $K(a, n) \equiv \omega^{|x|}$.*

We adapt Chen's theorem to our setting, which involves well-orderings instead of notations. Our proof is a direct adaptation of Shore's proof of Chen's theorem. We begin by defining some computable operations on trees.

Definition 2.20 (Shore [40, Definition 3.9], slightly modified)**.** For any (possibly finite) sequence of trees $\langle T_i \rangle$, we define their *maximum* by joining all T_i's at the root, i.e.,

$$\max(\langle T_i \rangle) = \{\langle\rangle\} \cup \{i^\frown\sigma : \sigma \in T_i\}.$$

Next, we define the *minimum* of a sequence of trees to be their "staggered common descent tree". More precisely, for any (possibly finite) sequence of trees $\langle T_i \rangle$, a node at level n of the tree $\min(\langle T_i \rangle)$ consists of, for each $i < n$ such that T_i is defined, a chain in T_i of length n. A node extends another node if for each i in their common domain, the i^{th} chain in the former node is an end-extension of the i^{th} chain in the latter node.

It is easy to see that the maximum and minimum operations play well with the ranks of trees:

Lemma 2.21 (Shore [40, Lemma 3.10])**.** *Let $\langle T_i \rangle$ be a (possibly finite) sequence of trees.*

1. *If $\mathrm{rk}(T_i) \leq \alpha$ for all i, then $\mathrm{rk}(\max(\langle T_i \rangle)) \leq \alpha$.*
2. *If there is some i such that T_i is ill-founded, then $\max(\langle T_i \rangle)$ is ill-founded.*
3. *If every T_i is well-founded, then $\mathrm{rk}(\min(\langle T_i \rangle)) \leq \mathrm{rk}(T_i) + i$.*
4. *If every T_i is ill-founded, then $\min(\langle T_i \rangle)$ is ill-founded as well.*

With the maximum and minimum operations in hand, we may prove an analog of Theorem 3.11 in Shore [40]:

Theorem 2.22. *Given a labeled well-ordering $\mathcal{L}$, we can uniformly compute sequences of trees $\langle g(a,n) \rangle_{n \in \mathbb{N}, a \in L}$ and $\langle h(a,n) \rangle_{n \in \mathbb{N}, a \in L}$ such that:*

- *if $n \in Y_a$, then $\mathrm{rk}(g(a,n)) \leq \omega \cdot \mathrm{otp}(L \restriction a)$ and $h(a,n)$ is ill-founded;*
- *if $n \notin Y_a$, then $\mathrm{rk}(h(a,n)) \leq \omega \cdot \mathrm{otp}(L \restriction a)$ and $g(a,n)$ is ill-founded.*

Proof. We define g and h by $\mathcal{L}$-effective transfinite recursion on L. For the base case (recall $Y_{0_L} = \mathcal{L}$), define $g(0_L, n)$ to be an infinite path of 0's for all $n \notin \mathcal{L}$, and the empty node for all $n \in \mathcal{L}$. Define $h(0_L, n)$ analogously.

For b limit, define $g(b, \langle a, n\rangle) = g(a, n)$ and $h(b, \langle a, n\rangle) = h(a, n)$ for any $n \in \mathbb{N}$ and $a <_L b$.

For $b = a + 1$, fix a Turing functional W which computes X from X' for any X. In particular,

$$n \in Y_b \quad \text{iff} \quad (\exists \langle P, Q, n\rangle \in W)(P \subseteq Y_a \text{ and } Q \subseteq Y_a^c).$$

Then define

$$h(b, n) = \max(\langle \min(\langle \{h(a, p) : p \in P\}, \{g(a, q) : q \in Q\}\rangle) : \langle P, Q, n\rangle \in W\rangle).$$

If $n \in Y_b$, then there is some $\langle P, Q, n\rangle \in W$ such that $P \subseteq Y_a$ and $Q \subseteq Y_a^c$. Then every tree in the above minimum for $\langle P, Q, n\rangle$ is ill-founded, so the minimum is itself ill-founded. Hence $h(b, n)$ is ill-founded.

If $n \notin Y_b$, then for all $\langle P, Q, n\rangle \in W$, either $P \nsubseteq Y_a$ or $Q \nsubseteq Y_a^c$. Either way, all of the above minima have rank $< \omega \cdot \text{otp}(L \upharpoonright a) + \omega$. Hence $h(b, n)$ has rank at most $\omega \cdot \text{otp}(L \upharpoonright a) + \omega \leq \omega \cdot \text{otp}(L \upharpoonright b)$.

Similarly, define

$$g(b, n) = \min(\langle \max(\langle \{g(a, p) : p \in P\}, \{h(a, q) : q \in Q\}\rangle) : \langle P, Q, n\rangle \in W\rangle).$$

This completes the construction for the successor case. □

Next, we adapt the above construction to obtain well-founded trees. To that end, for each well-ordering L, we aim to compute a tree $(T(\omega \cdot L))^\infty$ which is universal for all trees of rank $\leq \omega \cdot \text{otp}(L)$. Shore [40, Definition 3.12] constructs such a tree by effective transfinite recursion. Instead, we use a simpler construction of Greenberg and Montalbán [20].

Definition 2.23. Given a linear ordering L, define $T(L)$ to be the tree of finite $<_L$-decreasing sequences, ordered by extension.

It is easy to see that L is well-founded if and only if $T(L)$ is well-founded, and if L is well-founded, then $\mathrm{rk}(T(L)) = \mathrm{otp}(L)$.

Definition 2.24 ([20, Definition 3.20])**.** Given a tree T, define a tree

$$T^\infty = \{\langle(\sigma_0, n_0), \dots, (\sigma_k, n_k)\rangle : \langle\rangle \neq \sigma_0 \subsetneq \cdots \subsetneq \sigma_k \in T, n_0, \dots, n_k \in \mathbb{N}\},$$

ordered by extension.

Lemma 2.25 ([20, §3.2.2])**.** *Let T be well-founded. Then*

1. *T^∞ is well-founded and $\mathrm{rk}(T^\infty) = \mathrm{rk}(T)$;*
2. *for every $\sigma \in T^\infty$ and $\gamma < \mathrm{rk}_{T^\infty}(\sigma)$, there are infinitely many immediate successors τ of σ in T^∞ such that $\mathrm{rk}_{T^\infty}(\tau) = \gamma$;*
3. *$\mathrm{KB}(T)$ embeds into $\mathrm{KB}(T^\infty)$;*
4. *$\mathrm{KB}(T^\infty) \equiv \omega^{\mathrm{rk}(T)} + 1$, hence $\mathrm{KB}(T^\infty) - \{\emptyset\}$ is indecomposable.*
5. *if S is well-founded and $\mathrm{rk}(S) \leq \mathrm{rk}(T)$ ($\mathrm{rk}(S) < \mathrm{rk}(T)$ resp.), then $\mathrm{KB}(S)$ embeds (strictly resp.) into $\mathrm{KB}(T^\infty)$.*

Here, $\mathrm{KB}(T)$ denotes the Kleene-Brouwer ordering restricted to T (Definition 1.13).

Proof. (3) and (5) are not stated in [20], so we give proofs. By (1), fix a rank function $r : T \to \mathrm{rk}(T^\infty) + 1$. We construct an embedding $f : T \to T^\infty$ which preserves rank (i.e., $r(\sigma) = \mathrm{rk}_{T^\infty}(f(\sigma))$), $<_{\mathrm{KB}}$, and level. Start by defining $f(\emptyset) = \emptyset$. Note that $r(\emptyset) = \mathrm{rk}(T^\infty) = \mathrm{rk}_{T^\infty}(\emptyset)$.

Suppose we have defined f on $\sigma \in T$. Then, we extend f by mapping each immediate successor τ of σ to an immediate successor $f(\tau)$ of $f(\sigma)$ such that $r(\tau) = \mathrm{rk}_{T^\infty}(f(\tau))$. Such $f(\tau)$ exists by (2). Furthermore, by (2), if we start defining f from the leftmost immediate successor of σ and proceed to the right, we can extend f in a way that preserves $<_{\mathrm{KB}}$. This proves (3).

(5) follows from (3) applied to S and (4) applied to S and T. □

Finally, we prove our analog of Chen's theorem (Theorem 2.19):

Theorem 2.26. *Given a labeled well-ordering $\mathcal{L}$, we can uniformly compute an indecomposable well-ordering M and well-orderings $\langle K(a,n)\rangle_{n\in\mathbb{N},a\in L}$ such that:*

- *if $n \in Y_a$, then $K(a,n) \equiv M$.*
- *if $n \notin Y_a$, then $K(a,n) < M$.*

Proof. Given $\mathcal{L}$, we may use Theorem 2.22, Definition 2.23 and Definition 2.24 to uniformly compute

$$M = \mathrm{KB}(T(\omega \cdot L)^\infty) - \{\emptyset\}$$

$$K(a,n) = \mathrm{KB}(\min\{T(\omega \cdot L)^\infty, h(a,n)\}) - \{\emptyset\} \qquad \text{for } n \in \mathbb{N}, a \in L.$$

By Lemma 2.25(4), M is indecomposable. Also,

$$\mathrm{rk}(T(\omega \cdot L)^\infty) = \omega \cdot \mathrm{otp}(L)$$

$$\text{so} \qquad \mathrm{rk}(\min\{T(\omega \cdot L)^\infty, h(a,n)\}) \leq \omega \cdot \mathrm{otp}(L).$$

It then follows from Lemma 2.25(5) that $K(a,n) \leq M$.

If $n \in Y_a$, then $h(a,n)$ is ill-founded. Fix some descending sequence $\langle \sigma_i \rangle_i$ in $h(a,n)$. Then we may embed $T(\omega \cdot L)^\infty$ into

$\min\{T(\omega \cdot L)^\infty, h(a,n)\}$ while preserving $<_{\mathrm{KB}}$: map τ to $\langle\langle \tau \restriction i, \sigma_i\rangle\rangle_{i=0}^{|\tau|}$. Therefore $M \leq K(a,n)$, showing that $K(a,n) \equiv M$ in this case.

If $n \notin Y_a$, then $\mathrm{rk}(h(a,n)) \leq \omega \cdot \mathrm{otp}(L \restriction a)$. Therefore

$$\mathrm{rk}(\min\{T(\omega \cdot L)^\infty, h(a,n)\}) \leq \omega \cdot \mathrm{otp}(L \restriction a) + 1.$$

Since $\omega \cdot \mathrm{otp}(L \restriction a) + 1 < \omega \cdot \mathrm{otp}(L)$, by Lemma 2.25(5), $K(a,n) < M$. □

2.5 Reducing ATR to WCWO

In this section, we apply Theorem 2.26 to show that ATR $\leq_W$ WCWO (Theorem 2.30). Together with Proposition 2.15, that implies that WCWO $\equiv_W$ CWO $\equiv_W$ ATR.

First we work towards some sort of modulus for jump hierarchies. The next two results are adapted from Shore [40, Theorem 2.3]. We have added uniformities where we need them.

Proposition 2.27. *Given a labeled well-ordering $\mathcal{L}$ and $a \in L$, we can uniformly compute an index for a $\Pi_1^{0,\mathcal{L}}$-singleton $\{f\}$ which is strictly increasing, and Turing reductions witnessing that $f \equiv_T Y_a$.*

Proof. By $\mathcal{L}$-effective transfinite recursion on L, we can compute an index for Y_a as a $\Pi_2^{0,\mathcal{L}}$-singleton (see Sacks [39, Proposition II.4.1]). Define f to be the join of Y_a and the lex-minimal Skolem function which witnesses that Y_a satisfies the $\Pi_2^{0,\mathcal{L}}$ predicate that we computed. Then we can compute an index for f as a $\Pi_1^{0,\mathcal{L}}$-singleton (see Jockusch, McLaughlin [27, Theorem 3.1]). Clearly we can compute Turing reductions witnessing that $Y_a \leq_T f \leq_T \mathcal{L} \oplus Y_a$. Next, we can $\mathcal{L}$-uniformly compute a Turing reduction from $Y_{0_L} = \mathcal{L}$ to Y_a, and hence a Turing reduction from $\mathcal{L} \oplus Y_a$ to Y_a.

Finally, without loss of generality, we can replace $f : \mathbb{N} \to \mathbb{N}$ with the strictly increasing function $n \mapsto \sum_{m \leq n}(f(m) + 1)$. □

Definition 2.28. For any $f, g : \mathbb{N} \to \mathbb{N}$, we say that g *majorizes* f if for all n, $g(n) \geq f(n)$. We say that g *dominates* f if for all sufficiently large n, $g(n) \geq f(n)$.

Lemma 2.29. *There are indices e_0, e_1, and e_2 such that for all labeled well-orderings $\mathcal{L}$ and $a \in L$, there is some strictly increasing $f : \mathbb{N} \to \mathbb{N}$ such that if Y_a is the a^{th} column of the unique hierarchy on L, then:*

1. *$\Phi_{e_0}^{\mathcal{L} \oplus a}$ is an index for a Turing reduction from f to Y_a;*
2. *for all $g : \mathbb{N} \to \mathbb{N}$, $\Phi_{e_1}^{\mathcal{L} \oplus a \oplus g}(0)\downarrow$ if and only if g does not majorize f;*
3. *for all g which majorizes f, $\Phi_{e_2}^{\mathcal{L} \oplus a \oplus g}$ is total and defines Y_a.*

Proof. Given $\mathcal{L}$ and $a \in L$, first use Proposition 2.27 to compute a tree T with a unique path f which is strictly increasing, and Turing reductions witnessing that $f \equiv_T Y_a$. This shows (1).

Given $g : \mathbb{N} \to \mathbb{N}$, we can compute the g-bounded subtree T_g of T. If g does not majorize f, then T_g has no infinite path. In that case, T_g is finite by König's lemma, hence we can eventually enumerate that fact. This shows (2).

If g majorizes f, then we can compute f as follows: $\sigma \prec f$ if and only if for all other τ with $|\tau| = |\sigma|$, the g-bounded subtree of T above τ is finite. We can then compute Y_a from f. This shows (3). □

We now combine Theorem 2.26 with the above lemma to prove that

Theorem 2.30. $\mathsf{ATR} \leq_W \mathsf{WCWO}$.

Proof. We reduce the version of ATR in Proposition 2.8 to WCWO. Given a labeled well-ordering $\mathcal{L}$ and $a \in L$, by Lemma 2.29, there is some strictly increasing f such that if g majorizes f, then $\mathcal{L} \oplus a \oplus g$ uniformly computes Y_a.

Furthermore, we may compute reductions witnessing $\text{range}(f) \leq_T f \leq_T Y_a$. From that we may compute a many-one reduction r from $\text{range}(f)$ to Y_{a+1} (the $(a+1)^{\text{th}}$ column of the unique hierarchy on $(L \upharpoonright \{b : b \leq_L a\}) + 1$).

Next, use $\mathcal{L}$ to compute labels for $(L \upharpoonright \{b : b \leq_L a\}) + 1$. Apply Theorem 2.26 to $(L \upharpoonright \{b : b \leq_L a\}) + 1$ (and its labels) to compute an indecomposable well-ordering M and for each n, a well-ordering $L_n := K(a+1, r(n))$, such that

$$\begin{aligned} n \in \text{range}(f) \quad &\Leftrightarrow \quad r(n) \in Y_{a+1} \quad \Leftrightarrow \quad L_n \equiv M \\ n \notin \text{range}(f) \quad &\Leftrightarrow \quad r(n) \notin Y_{a+1} \quad \Leftrightarrow \quad L_n < M. \end{aligned}$$

For the forward functional, consider the following WCWO-instance:

$$\sum_n M \quad \text{and} \quad \left(\sum_n L_n\right) + 1.$$

Observe that by Lemma 2.17, $\sum_n L_n$ has the same ordertype as $\sum_n M$. Hence any WCWO-solution F must go from left to right. Furthermore, since M is indecomposable, it has no last element, so F must embed $\sum_n M$ into $\sum_n L_n$.

For the backward functional, we start by uniformly computing any element m_0 of M. Then we use F to compute the following function:

$$g(n) = \pi_0(F(\langle n+1, m_0 \rangle)).$$

We show that g majorizes f. For each n, F embeds $M \cdot n$ into $\sum_{i \leq g(n)} L_i$. It follows from Lemma 2.18 that at least n of the L_i's ($i \leq g(n)$) must have ordertype M. That means that there must be at least n elements in the range of f which lie below $g(n)$, i.e., $f(n) \leq g(n)$.

Since g majorizes f, $\mathcal{L} \oplus a \oplus g$ uniformly computes Y_a by Lemma 2.29, as desired. □

Using Theorem 2.30 and Proposition 2.15, we conclude that

Corollary 2.31. *$\mathsf{CWO} \equiv_W \mathsf{ATR} \equiv_W \mathsf{WCWO}$.*

2.6 Reducing ATR to $\mathsf{NDS}_{\mathrm{WO}}$ and $\mathsf{NIAC}_{\mathrm{WO}}$

Shore [40, Theorem 3.7] showed that in reverse mathematics, $\mathsf{NDS}_{\mathrm{WO}}$ (formulated as a Π^1_2 sentence) implies ATR_0 over RCA_0. We adapt his proof to show that

Theorem 2.32. *$\mathsf{ATR} \leq_W \mathsf{C}_{\mathbb{N}} * \mathsf{NDS}_{\mathrm{WO}}$. In particular, $\mathsf{ATR} \leq_c \mathsf{NDS}_{\mathrm{WO}}$ and $\mathsf{ATR} \leq_W^{\mathrm{arith}} \mathsf{NDS}_{\mathrm{WO}}$.*

Proof. We reduce the version of ATR in Proposition 2.8 to $\mathsf{NDS}_{\mathrm{WO}}$. Given a labeled well-ordering $\mathcal{L}$ and $a \in L$, by Lemma 2.29, there is some strictly increasing f such that if g majorizes f, then $\mathcal{L} \oplus a \oplus g$ uniformly computes Y_a. Furthermore, as in the proof of Theorem 2.30, we may compute a many-one reduction r from f to Y_{a+1}.

Next, use $\mathcal{L}$ to compute labels for $(L \upharpoonright \{b : b \leq_L a\}) + 1$. Apply Theorem 2.26 to $(L \upharpoonright \{b : b \leq_L a\}) + 1$ to compute an indecomposable well-ordering M and for each i and n, a well-ordering $K(a+1, r(i,n))$, such that

$$f(i) = n \quad \Leftrightarrow \quad r(i,n) \in Y_{a+1} \quad \Leftrightarrow \quad K(a+1, r(i,n)) \equiv M$$
$$f(i) \neq n \quad \Leftrightarrow \quad r(i,n) \notin Y_{a+1} \quad \Leftrightarrow \quad K(a+1, r(i,n)) < M.$$

For the forward functional, define for each j and n:

$$L_{j,n} = \sum_{j \le i < n} K(a+1, r(i,n))$$
$$N_j = \sum_n L_{j,n}.$$

For each j and n, $L_{j+1,n}$ uniformly embeds into $L_{j,n}$. So for each j, we can uniformly embed N_{j+1} into N_j. Hence $\langle N_j \rangle_j$ (with said embeddings) is an $\mathsf{NDS}_{\mathrm{WO}}$-instance.

Apply $\mathsf{NDS}_{\mathrm{WO}}$ to obtain some embedding $F : N_j \to N_k$, $j < k$. For the backward functional, we aim to compute a sequence $\langle h_q \rangle_q$ of functions, such that h_q majorizes f for all sufficiently large q. We start by uniformly computing any element m_0 of M. Then for each q, define

$$h_q(0) = q \quad \text{and} \quad h_q(n+1) = \pi_0(F(\langle h_q(n)+1, m_0 \rangle)).$$

We show that $h_{f(k)}$ majorizes f. (Hence for all $q \ge f(k)$, h_q majorizes f.) For this proof, temporarily set $q = f(k)$. We show by induction on n that $h_q(n) \ge f(k+n)$. The base case $n = 0$ holds by definition of q.

Suppose $h_q(n) \ge f(k+n)$. For each $j \le i \le k+n$, $K(a+1, r(i, f(i)))$ is a summand in $L_{j,f(i)}$ (because $f(i) > i$), which is in turn a summand in $\sum_{m \le h_q(n)} L_{j,m}$. That implies that $M \cdot (k+n-j+1)$ embeds into $\sum_{m \le h_q(n)} L_{j,m}$, which lies below $\langle h_q(n)+1, m_0 \rangle$ in N_j.

Composing with F, we deduce that $M \cdot (k+n-j+1)$ embeds into the initial segment of N_k below $F(\langle h_q(n)+1, m_0 \rangle)$, which is contained in $\sum_{m \le h_q(n+1)} L_{k,m}$.

It follows from Lemma 2.18 that there are at least $k+n-j+1$ many copies of M in $\sum_{m \leq h_q(n+1)} L_{k,m}$. Therefore, there are at least $k+n-j+1$ many elements in $\{f(i) : i \geq k\}$ below $h_q(n+1)$. It follows that

$$h_q(n+1) \geq f(k+n-j+k) \geq f(k+n+1)$$

as desired. This completes the proof of the inductive step. We have shown that $h_{f(k)}$ majorizes f.

Finally, by Lemma 2.29(2), given $\mathcal{L} \oplus a \oplus \langle h_q \rangle_q$, we may apply $\mathsf{C}_{\mathbb{N}}$ (Definition 1.10) to compute some q such that h_q majorizes f. Then $\mathcal{L} \oplus a \oplus h_q$ uniformly computes Y_a by Lemma 2.29(3), as desired. □

The above proof can be easily modified to show that

Theorem 2.33. $\mathsf{ATR} \leq_W \mathsf{C}_{\mathbb{N}} * \mathsf{NIAC}_{\mathrm{WO}}$. *In particular,* $\mathsf{ATR} \leq_c \mathsf{NIAC}_{\mathrm{WO}}$ *and* $\mathsf{ATR} \leq_W^{\mathrm{arith}} \mathsf{NIAC}_{\mathrm{WO}}$.

Proof. Given $\mathcal{L}$ and $a \in L$, compute $\langle L_{j,n} \rangle_{j,n}$ and $\langle N_j \rangle_j$ as in the proof of Theorem 2.32. Then consider the $\mathsf{NIAC}_{\mathrm{WO}}$-instance $\langle N_j + j \rangle_j$.

Given an embedding $F : N_j + j \to N_k + k$, first observe that by Lemma 2.17, N_j and N_k have the same ordertype, namely that of $M \cdot \omega$. Hence $j < k$. Furthermore, since M is indecomposable, F must embed N_j into N_k. The backward functional is then identical to that in Theorem 2.32. □

We do not know if $\mathsf{ATR} \leq_W \mathsf{NDS}_{\mathrm{WO}}$, $\mathsf{ATR} \leq_W \mathsf{NIAC}_{\mathrm{WO}}$, or even $\mathsf{ATR} \leq_W \mathsf{WQO}_{\mathrm{WO}}$.

CHAPTER 3

KÖNIG'S DUALITY THEOREM AND TWO-SIDED PROBLEMS

In this chapter, we define several "two-sided" problems, which are natural extensions of some of the problems we studied in chapter 2. This allows us to calibrate the computational content of König's duality theorem for countable bipartite graphs. In particular, we define a two-sided version of ATR, denoted ATR_2 (Definition 3.2), and show that the problem of computing a König cover of a given bipartite graph is (roughly) as hard as ATR_2 (Theorems 3.39 and 3.41).

ATR_2 is much harder than ATR in terms of computational difficulty (Corollary 3.8), yet König's duality theorem is equivalent to ATR_0 in reverse mathematics (Aharoni, Magidor, Shore [2], Simpson [41]). Therefore, this exhibits a marked difference between computable reducibilities and reverse mathematics.

The two-sided problems we study and König's duality theorem also provide examples of problems which lie strictly between $\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$ and $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$ in the Weihrauch degrees. Other examples exhibiting similar phenomena were studied by Kihara, Marcone, Pauly [28].

3.1 Two-sided problems

Many of the problems we have considered thus far have domains which are Π^1_1. For instance, the domain of CWO is the set of pairs of well-orderings. In that case, being outside the domain is a Σ^1_1 property. Now, any Σ^1_1 property can be thought of as a problem whose instances are sets satisfying said property and solutions are sets which witness that said property holds. This suggests that we combine a problem which has a Π^1_1 domain with the problem corresponding to the complement of its domain.

One obvious way to combine such problems is to take their union. For example, a "two-sided" version of WCWO could map pairs of well-orderings to any embedding between them, and map other pairs of linear orderings to any infinite descending sequence in either linear ordering. We will not consider such problems here, because they are not Weihrauch reducible (or even arithmetically Weihrauch reducible) to $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$. (Any such reduction could be used to give a Σ^1_1 definition for the set of indices of pairs of well-orderings. See also Brattka, de Brecht, Pauly [5, Theorem 7.7].) On the other hand, it is not hard to see that the problems corresponding to Fraïssé's conjecture ($\mathsf{WQO}_{\mathrm{LO}}$) and König's duality theorem (see section 3.2) are Weihrauch reducible to $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$.

However, note that embeddings between linear orderings can still exist even when either linear ordering is ill-founded! This suggests an alternative method of combination, resulting in the following "two-sided" extensions of CWO and WCWO.

Definition 3.1. Define the following problems:

CWO_2: Given linear orderings L and M, either produce an embedding from one of them onto an initial segment of the other, or an infinite descending sequence in either ordering. In either case we indicate which type of solution we produce.

WCWO_2: Given linear orderings L and M, either produce an embedding from one of them into the other, or an infinite descending sequence in either ordering. In either case we indicate which type of solution we produce.

It is not hard to see that whether solutions to instances of the above problems come with an indication of their type does not affect the Weihrauch degree of the problems. Hence we include the type for our convenience.

Next, we define a two-sided version of ATR. In section 3.2, we will show that it is closely related to König's duality theorem (Theorem 3.39).

Recall our definition of a jump hierarchy:

Definition 2.2. Let L be a linear ordering with first element 0_L, and let $A \subseteq \mathbb{N}$. We say that $\langle X_a \rangle_{a \in L}$ is a *jump hierarchy on L which starts with A* if $X_0 = A$ and for all $b >_L 0_L$, $X_b = (\bigoplus_{a <_L b} X_a)'$.

Jump hierarchies on ill-founded linear orderings were first studied by Harrison [22], and are often called pseudohierarchies. See, for example, [42, Section V.4]).

Definition 3.2. We define a two-sided version of ATR as follows:

ATR_2: Given a linear ordering L and a set $A \subseteq \mathbb{N}$, either produce an infinite $<_L$-descending sequence S, or a jump hierarchy $\langle X_a \rangle_{a \in L}$ on L which begins with A. In either case we indicate which type of solution we produce.[1]

Just as for CWO and WCWO, if we require an ATR_2-solution to an ill-founded L to be an infinite $<_L$-descending sequence, then the resulting problem is not Weihrauch reducible to $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$. The same holds if we require an ATR_2-solution to L to be a jump hierarchy whenever L supports a jump hierarchy, because

Theorem 3.3 (Harrington, personal communication)**.** *The set of indices for linear orderings which support a jump hierarchy is Σ^1_1-complete.*

A Weihrauch reduction from the aforementioned variant of ATR_2 to $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$ would

[1]Just as for CWO_2 and WCWO_2, this does not affect the Weihrauch degree of ATR_2.

yield a Π^1_1 definition of the set of indices for linear orderings which support a jump hierarchy, contradicting Harrington's result.

Next, we determine the positions of CWO_2, WCWO_2, and ATR_2 relative to $\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$ and $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$ in the Weihrauch degrees. In addition, even though we are not viewing $\mathsf{WQO}_{\mathrm{LO}}$ (Fraïssé's conjecture) as a two-sided problem, most of our arguments and results hold for $\mathsf{WQO}_{\mathrm{LO}}$ as well.

First observe that each of CWO, WCWO, and ATR is trivially Weihrauch reducible to its two-sided version. By Corollary 2.31 and the fact that $\mathsf{ATR} \equiv_W \mathsf{UC}_{\mathbb{N}^\mathbb{N}}$ (Kihara, Marcone, Pauly [28]), these two-sided problems lie above $\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$ in the Weihrauch degrees. We do not know if $\mathsf{WQO}_{\mathrm{LO}}$ lies above $\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$ in the Weihrauch degrees.

Next observe that CWO_2, WCWO_2, ATR_2, and $\mathsf{WQO}_{\mathrm{LO}}$ are each defined by an arithmetic predicate on an arithmetic domain. It easily follows that they lie below $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$ in the Weihrauch degrees. In fact, they lie strictly below $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$:

Proposition 3.4. *Suppose that P is an arithmetically defined multivalued function such that* $\mathrm{dom}(P)$ *is not* Π^1_1. *If Q is arithmetically defined and* $\mathrm{dom}(Q)$ *is arithmetic, then P is not arithmetically Weihrauch reducible to Q.*

Proof. If P is arithmetically Weihrauch reducible to Q via arithmetically defined functionals Φ and Ψ, then we could give a Π^1_1 definition for $\mathrm{dom}(P)$ as follows: $X \in \mathrm{dom}(P)$ if and only if

$$\Phi(X) \in \mathrm{dom}(Q) \wedge \forall Y[Y \in Q(\Phi(X)) \to \Psi(X \oplus Y) \in P(X)].$$

Contradiction. □

Corollary 3.5. $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$ *is not arithmetically Weihrauch reducible to any of* CWO_2*,* WCWO_2*,* ATR_2*, or* $\mathsf{WQO}_{\mathrm{LO}}$*.*

Proof. Each of CWO_2, WCWO_2, ATR_2, and $\mathsf{WQO}_{\mathrm{LO}}$ are arithmetically defined with arithmetic domain. $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$ is also arithmetically defined, but its domain is Σ^1_1-complete. Apply Proposition 3.4. □

Next we show that CWO_2, WCWO_2, ATR_2, and $\mathsf{WQO}_{\mathrm{LO}}$ are not Weihrauch reducible (or even computably reducible) to $\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$. First we have a boundedness argument:

Lemma 3.6. *Suppose* $P(d, Y)$ *is a* Π^1_1 *predicate on* $\mathbb{N} \times \mathbb{N}^\mathbb{N}$ *and* D *is a* Σ^1_1 *subset of* $\mathbb{N}$*. If for every* $d \in D$*, there is some hyperarithmetic* Y *such that* $P(e, Y)$ *holds, then there is some* $\alpha < \omega_1^{CK}$ *such that for every* $d \in D$*,* $P(d, \cdot)$ *has a solution below level* α *of the hyperarithmetical hierarchy.*

Proof. Consider the following predicate of d and a:

$$d \notin D \vee (a \in W \wedge P(d, \cdot) \text{ has a solution below level } a).$$

This predicate is Π^1_1: assuming that $a \in W$, $P(d, \cdot)$ has a solution below level a if and only if there is some $e \in \mathbb{N}$ such that for all Y which is a jump hierarchy along a which starts with $\emptyset$, Φ^Y_e is total and $P(d, Y)$ holds.

By Π^1_1-uniformization (Theorem 1.16), there is a Π^1_1 predicate $Q(d, a)$ uniformizing P. Then the set

$$\{a : (\exists d \in D) Q(d, a)\} = \{a : (\exists d \in D)(\forall b \neq a) \neg Q(d, b)\}$$

is Σ^1_1 and contained in W. By boundedness (Theorem 1.14), it is bounded by some $\alpha < \omega_1^{CK}$, proving the desired statement. □

Corollary 3.7. *Each of* CWO_2, WCWO_2, ATR_2, *and* $\mathsf{WQO}_{\mathrm{LO}}$ *have a computable instance with no hyperarithmetic solution.*

Proof. By the contrapositive of Lemma 3.6, it suffices to show that for any $\alpha < \omega_1^{CK}$, there is a computable instance of each problem with no solution which lies below level α in the hyperarithmetical hierarchy.

Observe that for any $\alpha < \omega_1^{CK}$, there is a computable instance of ATR (take any computable well-ordering longer than α) such that its solution lies above level α in the hyperarithmetical hierarchy.[2] The following reductions imply that the same holds for WCWO_2, CWO_2, ATR_2, and $\mathsf{WQO}_{\mathrm{LO}}$:

$\mathsf{ATR} \leq_W \mathsf{WCWO} \leq_W \mathsf{WCWO}_2 \leq_W \mathsf{CWO}_2$	Theorem 2.30
$\mathsf{ATR} \leq_W \mathsf{ATR}_2$	
$\mathsf{ATR} \leq_c \mathsf{WQO}_{\mathrm{LO}}$	Theorem 2.32

This completes the proof. □

Corollary 3.7 implies that

Corollary 3.8. CWO_2, WCWO_2, ATR_2, *and* $\mathsf{WQO}_{\mathrm{LO}}$ *are not computably reducible or arithmetically Weihrauch reducible to* $\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$.

We conclude that

Corollary 3.9. *Let* P *be* CWO_2, WCWO_2 *or* ATR_2. *Then*

$$\mathsf{UC}_{\mathbb{N}^\mathbb{N}} <_W P <_W \mathsf{C}_{\mathbb{N}^\mathbb{N}}.$$

In fact, $P \not\leq_W^{\mathrm{arith}} \mathsf{UC}_{\mathbb{N}^\mathbb{N}}$, $P \not\leq_c \mathsf{UC}_{\mathbb{N}^\mathbb{N}}$, *and* $\mathsf{C}_{\mathbb{N}^\mathbb{N}} \not\leq_W^{\mathrm{arith}} P$.

[2] Note that the domain of ATR is not Σ^1_1, so we cannot apply Lemma 3.6 to show that there is a computable instance of ATR with no hyperarithmetic solution. (The latter statement is clearly false.)

Corollary 3.10. $\mathsf{WQO}_{\mathrm{LO}}$ *has the following relationships with* $\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$ *and* $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$*:*

- $\mathsf{UC}_{\mathbb{N}^\mathbb{N}} <_c \mathsf{WQO}_{\mathrm{LO}}$. *Also,* $\mathsf{WQO}_{\mathrm{LO}} \not\leq_W^{\mathrm{arith}} \mathsf{UC}_{\mathbb{N}^\mathbb{N}}$.
- $\mathsf{WQO}_{\mathrm{LO}} <_W \mathsf{C}_{\mathbb{N}^\mathbb{N}}$*; in fact* $\mathsf{C}_{\mathbb{N}^\mathbb{N}} \not\leq_W^{\mathrm{arith}} \mathsf{WQO}_{\mathrm{LO}}$.

3.1.1 ATR_2 and variants thereof

In this section, we prove some results regarding ATR_2 and its variants. First, we have several results showing that ATR_2 is fairly robust. At the end, we show that $\mathsf{CWO}_2 \leq_W \mathsf{ATR}_2$ (Theorem 3.14), in analogy with $\mathsf{CWO} \leq_W \mathsf{ATR}$ (Proposition 2.15).

We start with the following analog of Proposition 2.4:

Proposition 3.11. ATR_2 *is Weihrauch equivalent to the following problem. Instances are triples* (L, A, Θ) *where* L *is a linear ordering,* $A \subseteq \mathbb{N}$*, and* $\Theta(n, Y, A)$ *is an arithmetical formula whose only free variables are* n*,* Y *and* A*. Solutions are either infinite* $<_L$*-descending sequences, or hierarchies* $\langle Y_a \rangle_{a \in L}$ *such that for all* $b \in L$*,* $Y_b = \{n : \Theta(n, \bigoplus_{a <_L b} Y_a, A)\}$*. (As usual, solutions come with an indication of their type.)*

Proof. Roughly speaking, we extend the reductions defined in Proposition 2.4. First, ATR_2 is Weihrauch reducible to the above problem: for the forward reduction, given (L, A), consider (L, A, Θ) where $\Theta(n, Y, A)$ holds if either $Y = \emptyset$ and $n \in A$, or $n \in Y'$. The backward reduction is the identity.

Conversely, given (L, A, Θ), let k be one greater than the number of quantifier alternations in Θ. Apply ATR_2 to $(1 + k \cdot L + 2, L \oplus A)$. If we obtain an infinite descending sequence in $1+k\cdot L+2$, we can uniformly compute an infinite descending sequence in L and output that.

Otherwise, we obtain a jump hierarchy $\langle X_\alpha\rangle_{\alpha\in 1+k\cdot L+2}$. We want to use it to either compute a hierarchy on L, or an infinite $<_L$-descending sequence.

We start by using the recursion theorem to compute a $\langle X_{(a,k-1)}\rangle_{a\in L}$-partial recursive function $f : L \to \mathbb{N}$, as described in the proof of Proposition 2.4. Note that f may not be total.

Next, we compute $(\langle X_{(a,k-1)}\rangle_{a\in L})''$ and use that to decide whether f is total. If so, following the proof of Proposition 2.4, we may compute a hierarchy on L with the desired properties.

If not, we use $(\langle X_{(a,k-1)}\rangle_{a\in L})''$ to compute the complement of the domain of f in L. This set has no $<_L$-least element, by construction of f. Therefore, we can uniformly compute an infinite $<_L$-descending sequence within it. $\square$

Just as we defined labeled well-orderings, we may also define labeled linear orderings if said linear orderings have first elements. Then we have the following analog of Proposition 2.6:

Proposition 3.12. ATR_2 *is Weihrauch equivalent to the following problem: an instance is a labeled linear ordering* $\mathcal{L}$ *and a set* $A \subseteq \mathbb{N}$*, and a solution is an* ATR_2*-solution to* (L, A).

Proof. It suffices to reduce ATR_2 to the given problem. Given (L, A), we start by computing $\omega\cdot(1+L)$ and labels for it. Then we apply the given problem to $\omega\cdot(1+L)$ (and its labels) and the set $L\oplus A$.

If we obtain an infinite descending sequence in $\omega \cdot (1+L)$, we can uniformly compute an infinite descending sequence in L and output that.

Otherwise, we obtain a jump hierarchy $\langle X_{(n,\alpha)}\rangle_{n\in\omega,\alpha\in 1+L}$ which starts with $L \oplus A$. First use this hierarchy to compute L'', which tells us whether L has a first element. If not, we can uniformly compute an infinite descending sequence in L and output that.

Otherwise, we use the recursion theorem to compute a partial $\langle X_{(0,b)}\rangle_{b\in L}$-recursive function $f : L \to \mathbb{N}$, as described in the proof of Proposition 2.6. Then we compute

$$S = \left\{ b \in L : \langle \Phi^{X_{(0,a)}}_{f(a)} \rangle_{a <_L b} \text{ defines a jump hierarchy} \right\}$$

and consider two cases.

Case 1. If S is all of L, then we output $\langle \Phi^{X_{(0,a)}}_{f(a)} \rangle_{a\in L}$, which is a jump hierarchy on L which starts with A.

Case 2. Otherwise, observe that by construction of f, $L \backslash S$ has no $<_L$-least element. Then we can compute an infinite $<_L$-descending sequence in $L\backslash S$ and output that.

Finally, note that $\langle X_{(n,\alpha)}\rangle_{n\in\omega,\alpha\in 1+L}$ can compute the above case division and the output in each case. □

Proposition 3.12 will be useful in section 3.2. Using similar ideas, we can show that

Proposition 3.13. ATR_2 *is arithmetically Weihrauch equivalent to the following problem: an instance is a linear ordering L and a set $A \subseteq \mathbb{N}$, and a solution* *is an infinite* $<_L$*-descending sequence, or some* $\langle X_a \rangle_{a \in L}$ *such that* $X_{0_L} = A$ *and* $X'_a \leq_T X_b$ *for all* $0_L \leq_L a <_L b$.

Proof. It suffices to construct an arithmetic Weihrauch reduction from ATR_2 to the given problem. Given (L, A), the forward functional outputs $(L, L \oplus A)$. To define the backward functional: if the above problem gives us some infinite $<_L$-descending sequence then we output that. Otherwise, suppose we are given $\langle X_a \rangle_{a \in L}$ such that $X_{0_L} = A$ and $X'_a \leq_T X_b$ for all $0_L \leq_L a <_L b$.

We start by attempting to use $(\langle X_a \rangle_{a \in L})'''$-effective transfinite recursion along L to define a partial $(\langle X_a \rangle_{a \in L})'''$-recursive function $f : L \to \mathbb{N}$ such that $\langle \Phi^{X_a}_{f(a)} \rangle_{a \in L}$ is a jump hierarchy on L which starts with A.

For the base case, we use $X_{0_L} = L \oplus A$ to uniformly compute A. For $b >_L 0_L$, first use $(\bigoplus_{a \leq_L b} X_a)'''$ to find Turing reductions (for each $a <_L b$) witnessing that $X'_a \leq_T X_b$. Then we can use X_b to compute $(\bigoplus_{a <_L b} \Phi^{X_a}_{f(a)})'$. This completes the definition of f.

Next, compute

$$S = \left\{ b \in L : \langle \Phi^{X_a}_{f(a)} \rangle_{a <_L b} \text{ defines a jump hierarchy} \right\}$$

and consider two cases.

Case 1. If S is all of L, then we output $\langle \Phi^{X_a}_{f(a)} \rangle_{a \in L}$, which is a jump hierarchy on L which starts with A.

Case 2. Otherwise, observe that by construction of f, $L \backslash S$ has no $<_L$-least element. Then we can compute an infinite $<_L$-descending sequence in $L \backslash S$ and output that.

Finally, note that by choosing n sufficiently large, $(\langle X_a\rangle_{a\in L})^{(n)}$ can compute the above case division and the output in each case. $\square$

Next, in analogy with $\mathsf{CWO} \leq_W \mathsf{ATR}$ (Proposition 2.15), we have that

Theorem 3.14. $\mathsf{CWO}_2 \leq_W \mathsf{ATR}_2$.

Proof. Given linear orderings (L, M), define N by adding a first element 0_N and a last element m_N to L. Apply ATR_2 to the linear ordering N and the set $L \oplus M$. If we obtain an infinite descending sequence in N, we can use that to uniformly compute an infinite descending sequence in L.

Otherwise, using Proposition 3.11, we may assume that we obtain a hierarchy $\langle X_a\rangle_{a\in N}$ such that:

- $X_{0_N} = L \oplus M$;
- for all $b >_N 0_N$, $X_b = \left(\bigoplus_{a<_N b} X_a\right)'''$.

We start by attempting to use $\langle X_a\rangle_{a\in L}$-effective transfinite recursion along L to define a partial $\langle X_a\rangle_{a\in L}$-recursive function $f : L \to \mathbb{N}$ such that $\{(a, \Phi^{X_a}_{f(a)}(0)) \in L \times M : \Phi^{X_a}_{f(a)}(0)\downarrow\}$ is an embedding of an initial segment of L into an initial segment of M.

To define f, if we are given any $b \in L$ and $f \upharpoonright \{a : a <_L b\}$, we need to define $f(b)$, specifically $\Phi^{X_b}_{f(b)}(0)$. First use $X_b = (\bigoplus_{a<_L b} X_a)'''$ to compute whether all of the following hold:

1. for all $a <_L b$, $\Phi^{X_a}_{f(a)}(0)$ converges and outputs some element of M;
2. $\{\Phi^{X_a}_{f(a)}(0) : a <_L b\}$ is an initial segment of M;

3. there is an M-least element above $\{\Phi^{X_a}_{f(a)}(0) : a <_L b\}$.

If so, we output said M-least element; otherwise diverge. This completes the definition of $\Phi^{X_b}_{f(b)}(0)$.

Apply the recursion theorem to the definition above to obtain a partial $\langle X_a \rangle_{a \in L}$-recursive function $f : L \to \mathbb{N}$. Now, to complete the definition of the backward reduction we consider the following cases.

Case 1. f is total. Then following the proof of Proposition 2.15, we output $\{(a, \Phi^{X_a}_{f(a)}(0)) : a \in L\}$, which is an embedding from L onto an initial segment of M.

Case 2. There is no L-least element above $\{a \in L : \Phi^{X_a}_{f(a)}(0)\downarrow\}$. Then we can output an infinite L-descending sequence above $\{a \in L : \Phi^{X_a}_{f(a)}(0)\downarrow\}$.

Case 3. $\{\Phi^{X_a}_{f(a)}(0) : a \in L, \Phi^{X_a}_{f(a)}(0)\downarrow\} = M$. Then following the proof of Proposition 2.15, we output $\{(\Phi^{X_a}_{f(a)}(0), a) : a \in L, \Phi^{X_a}_{f(a)}(0)\downarrow\}$, which is an embedding from M onto an initial segment of L.

Case 4. There is no M-least element above $\{\Phi^{X_a}_{f(a)}(0) : a \in L, \Phi^{X_a}_{f(a)}(0)\downarrow\}$. Then we can output an infinite M-descending sequence which lies above $\{\Phi^{X_a}_{f(a)}(0) : a \in L, \Phi^{X_a}_{f(a)}(0)\downarrow\}$.

Finally, note that the last column X_{m_N} of $\langle X_a \rangle_{a \in N}$ can compute the above case division and the appropriate output for each case. □

3.2 König's duality theorem

In this section, we study König's duality theorem from the point of view of computable reducibilities.

First we state some definitions from graph theory. A graph G is *bipartite* if its vertex set can be partitioned into two sets such that all edges in G go from one of the sets to the other. It is not hard to see that G is bipartite if and only if it has no odd cycle. (Hence the property of being bipartite is Π^0_1.) A *matching* in a graph is a set of edges which are vertex-disjoint. A *(vertex) cover* in a graph is a set of vertices which contains at least one endpoint from every edge. König's duality theorem states that:

Theorem 3.15. *For any bipartite graph G, there is a matching M and a cover C which are* dual*, i.e., C is obtained by choosing exactly one vertex from each edge in M. Such a pair (C, M) is said to be a* König cover*.*

König proved the above theorem for finite graphs, where it is commonly stated as "the maximum size of a matching is equal to the minimum size of a cover". For infinite graphs, this latter form would have little value. Instead of merely asserting the existence of a bijection, we want such a bijection to respect the structure of the graph. Hence the notion of a König cover. Podewski and Steffens [37] proved König's duality theorem for countable graphs. Finally, Aharoni [1] proved it for graphs of arbitrary cardinality. In this thesis, we will only study the theorem for countable graphs.

Definition 3.16. KDT is the following problem: given a (countable) bipartite graph G, produce a König cover (C, M).

Note that we represent bipartite graphs as their vertex set and edge relation. Alternatively, our representation of a bipartite graph could also include a partition of its vertex set which witnesses that the graph is bipartite. Even though these two representations are not computably equivalent[3], all of our results hold for either representation.

Aharoni, Magidor, Shore [2] studied König's duality theorem for countable graphs from the point of view of reverse mathematics. They showed that ATR_0 is provable from König's duality theorem. They also showed that König's duality theorem is provable in the system Π^1_1-CA_0, which is strictly stronger than ATR_0. Simpson [41] then closed the gap by showing that König's duality theorem is provable in (hence equivalent to) ATR_0.

The proof of ATR_0 from König's duality theorem in [2] easily translates into a Weihrauch reduction from ATR to KDT. We adapt their proof to show that ATR_2 is Weihrauch reducible to $\mathsf{LPO} * \mathsf{KDT}$ (Theorem 3.39). Next, we adapt [41]'s proof of König's duality theorem from ATR_0 to show that KDT is arithmetically Weihrauch reducible to ATR_2 (Theorem 3.41). It follows that ATR_2 and KDT are arithmetically Weihrauch equivalent. Since both ATR_2 and KDT have computational difficulty far above the arithmetic (see, for example, Corollary 3.7), this shows that ATR_2 and KDT have roughly the same computational difficulty.

Before constructing the above reductions, we make some easy observations about KDT.

Proposition 3.17. $\mathsf{KDT} \leq_W \mathsf{C}_{\mathbb{N}^\mathbb{N}}$*, but* $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$ *is not even arithmetically Weihrauch reducible to* KDT*.*

[3]In fact, there is a computable bipartite graph such that no computable partition of its vertices witnesses that the graph is bipartite. This was known to Bean [4, remarks after Theorem 7] (we thank Jeff Hirst for pointing this out.) See also Hirst [26, Corollary 3.17].

Proof. The first statement holds because KDT is defined by an arithmetic predicate on an arithmetic domain. The second statement follows from Proposition 3.4. $\square$

Proposition 3.18. KDT *is parallelizable, i.e.,* $\widehat{\mathsf{KDT}} \leq_W \mathsf{KDT}$*.*

Proof. This holds because the disjoint union of bipartite graphs is bipartite, and any König cover of a disjoint union of graphs restricts to a König cover on each graph. $\square$

We do not know if ATR_2 is parallelizable; a negative answer would separate ATR_2 and KDT up to Weihrauch reducibility.

Since being a bipartite graph is a Π^0_1 property (in particular Π^1_1), we could define *two-sided* KDT (KDT_2): given a graph, produce an odd cycle (witnessing that the given graph is not bipartite) or a König cover. This produces a problem which is Weihrauch equivalent to KDT, however:

Proposition 3.19. $\mathsf{KDT}_2 \leq_W \mathsf{LPO} \times \mathsf{KDT}$*, hence* $\mathsf{KDT} \equiv_W \mathsf{KDT}_2$*.*

Proof. Given a KDT_2-instance G (i.e., a graph), we can uniformly compute a graph H which is always bipartite and is equal to G if G is bipartite: H has the same vertices as G, but as we enumerate edges of G into H, we omit any edges that would result in an odd cycle in the graph we have enumerated thus far.

For the reduction, we apply $\mathsf{LPO} \times \mathsf{KDT}$ to (G, H). If LPO (Definition 1.10) tells us that G is bipartite, we output a KDT-solution to $H = G$. Otherwise, we can uniformly compute and output an odd cycle in G.

Finally, to conclude that $\mathsf{KDT} \equiv_W \mathsf{KDT}_2$, we use Proposition 3.18 and the fact that $\mathsf{LPO} \leq_W \mathsf{KDT}$, which trivially follows from Theorem 3.33 later. $\square$

3.2.1 Reducing ATR_2 to KDT

For both of our forward reductions (from ATR or ATR_2 to KDT), the bipartite graphs we construct are sequences of subtrees of $\mathbb{N}^{<\mathbb{N}}$. Let us define our notation regarding trees. For us, a *rooted subtree* of $\mathbb{N}^{<\mathbb{N}}$ is a subset T of $\mathbb{N}^{<\mathbb{N}}$ for which there is a unique $r \in T$ (called the *root*) such that:

- no proper prefixes of r lie in T;
- for every $s \in T$, s extends r and every prefix of s which extends r lies in T.

A rooted subtree of $\mathbb{N}^{<\mathbb{N}}$ whose root is the empty node $\langle\rangle$ is just a prefix-closed subset of $\mathbb{N}^{<\mathbb{N}}$.

If $r \in \mathbb{N}^{<\mathbb{N}}$ and $R \subseteq \mathbb{N}^{<\mathbb{N}}$, we define $r^\frown R = \{r^\frown s : s \in R\}$. In particular, if $T \subseteq \mathbb{N}^{<\mathbb{N}}$ is prefix-closed, then $r^\frown T$ is a subtree of $\mathbb{N}^{<\mathbb{N}}$ with root r. Conversely, if a rooted subtree of $\mathbb{N}^{<\mathbb{N}}$ has root r, it is equal to $r^\frown T$ for some such T. If T is prefix-closed, we sometimes refer to a tree of the form $r^\frown T$ as a *copy* of T. (Our usage of "copy" is more restrictive than its usage in computable structure theory.)

If T is a rooted subtree of $\mathbb{N}^{<\mathbb{N}}$, for any $t \in T$, the *subtree of T above t* is the subtree $\{s \in T : t \preceq s\}$ with root t.

For each $r \in \mathbb{N}^{<\mathbb{N}}$, $e \in \mathbb{N}$ and $X \subseteq \mathbb{N}$, (r, e, X) is a name for the following tree T with root node r: $r^\frown\sigma \in T$ if and only if for all $k < |\sigma|$, $\Phi^X_{e,\prod_{i<k}(\sigma(i)+1)}(\sigma \upharpoonright k)\downarrow = 1$. This representation is easily seen to be computably equivalent to what is perhaps the usual representation, where if Φ^X_e is total, then (r, e, X) is the name for the tree defined by Φ^X_e starting with root r. The advantage of our representation is that (r, e, X) names some tree even if Φ^X_e is partial, which will be useful when e is produced by the recursion theorem.

Using the above representation, we can define a representation for sequences of subtrees of $\mathbb{N}^{<\mathbb{N}}$: view (e, X) as $\langle(\langle n\rangle, e_n, X)\rangle_n$, where e_n is an X-index for $\Phi_e^X(n, \cdot)$. Observe that every (e, X) names some such sequence.

Henceforth, we will use "tree" as a shorthand for "rooted subtree of $\mathbb{N}^{<\mathbb{N}}$".

Next, we describe our backward reduction for $\mathsf{ATR} \leq_W \mathsf{KDT}$. It only uses the cover in a König cover and not the matching. First we define a coding mechanism:

Definition 3.20. Given a tree T (with root r) and a König cover (C, M) of T, we can decode the bit b, which is the Boolean value of $r \in C$. We say that (C, M) *codes* b.

More generally, given any sequence of trees $\langle T_n : n \in X\rangle$ (with roots r_n) and a König cover (C_n, M_n) for each T_n, we can uniformly decode the following set from the set $\langle(C_n, M_n)\rangle$:

$$A = \{n \in X : r_n \in C_n\}.$$

We say that $\langle(C_n, M_n)\rangle$ *codes* A.

A priori, different König covers of the same tree or sequence of trees can code different bits or sets respectively. A tree or sequence of trees is *good* if that cannot happen:

Definition 3.21. A tree T is *good* if its root r lies in C for every König cover (C, M) of T, or lies outside C for every König cover (C, M) of T. A sequence of trees $\langle T_n\rangle$ is *good* if every T_n is good. In other words, $\langle T_n\rangle$ is good if all of its König covers code the same set.

If $\langle T_n\rangle$ is good and every (equivalently, some) König cover of $\langle T_n\rangle$ codes A, we say that $\langle T_n\rangle$ *codes* A.

We will use this coding mechanism to define the backward reduction in $\mathsf{ATR} \leq_W \mathsf{KDT}$. Here we make a trivial but important observation: for any $s \in \mathbb{N}^{<\mathbb{N}}$ and any tree T, the König covers of T and the König covers of $s^\frown T$ are in obvious correspondence, which respects whichever bit is coded. Hence T is good if and only if $s^\frown T$ is good.

Next, we set up the machinery for our forward reductions. Aharoni, Magidor, and Shore's [2] proof of ATR_0 from KDT uses effective transfinite recursion along the given well-ordering to construct good trees which code complicated sets. The base case is as follows:

Lemma 3.22. *Given any $A \subseteq \mathbb{N}$, we can uniformly compute a sequence of trees $\langle T_n \rangle$ which codes A.*

Proof. The tree $\{\langle\rangle\}$ codes the bit 0. This is because any matching must be empty, hence any dual cover must be empty.

The tree $\{\langle\rangle, \langle 0\rangle, \langle 1\rangle\}$ codes the bit 1. This is because any matching must contain exactly one of the two edges. Hence any cover dual to that must consist of a single node. But the root node is the only node which would cover both edges.

By defining each T_n to be either of the above trees as appropriate, we obtain a sequence $\langle T_n \rangle$ which codes A. □

We may use this as the base case for our construction as well. As for the successor case, however, we want to extract extra information from the construction in [2]. The issue is that when reducing ATR_2 to KDT, "effective transfinite recursion" on ill-founded linear orderings may produce garbage. (Of particular concern is that the resulting trees may not be good.) Nevertheless, we may attempt it anyway. If we detect inconsistencies in the resulting trees and König covers (using the extra information we have extracted), then we may use them to compute an infinite descending sequence in the given linear ordering. Otherwise, we may decode the resulting König covers to produce a jump hierarchy.

In order to describe our construction in detail, we need to examine the construction in [2] closely. First we state a sufficient condition on a König cover of a tree and a node in said tree which ensures that the given König cover, when restricted to the subtree above the given node, remains a König cover. The set of all nodes satisfying the former condition form a subtree:

Definition 3.23. For any tree T (with root r) and any König cover (C, M) of T, define the subtree T^* (with root r):

$$T^* = \{t \in T : \forall s(r \prec s \preceq t \to (s \notin C \vee (s \upharpoonright (|s|-1), s) \notin M))\}.$$

The motivation behind the definition of T^* is as follows. Suppose (C, M) is a König cover of T. If $s \in C$ and $(s \upharpoonright (|s|-1), s) \in M$, then C restricted to the subtree of T above s would contain s, but M restricted to said subtree would not contain any edge with endpoint s. This means that the restriction of (C, M) to said subtree is not a König cover. Hence we define T^* to avoid this situation. According to [2, Lemma 4.5], this is the only situation we need to avoid.

When we use the notation T^*, the cover (C, M) will always be clear from context. Observe that T^* is uniformly computable from T and (C, M).

Lemma 3.24. *For any T and any König cover (C, M) of T, define T^* as above. Then for any $t \in T^*$, (C, M) restricts to a König cover of the subtree of T (not T^*!) above t.*

Proof. Proceed by induction on the level of t using [2, Lemma 4.5]. $\square$

Using Definition 3.23 and Lemma 3.24, we may easily show that:

Proposition 3.25. *Let (C, M) be a König cover of T. Suppose that $t \in T^*$. Let S denote the subtree of T above t. Then S^* is contained in T^*, where S^* is calculated using the restriction of (C, M) to S.*

Next, we define a computable operation on trees which forms the basis of the proofs of [2, Lemmas 4.9, 4.10].

Definition 3.26. Given a (possibly finite) sequence of trees $\langle T_i \rangle$, each with the empty node as root, we may *combine* it to form a single tree S, by adjoining two copies of each T_i to a root node r. Formally,

$$S = \{r\} \cup \{r^\frown(i, 0)^\frown\sigma : \sigma \in T_i\} \cup \{(i, 1)^\frown\sigma : \sigma \in T_i\}.$$

Logically, the combine operation can be thought of as $\neg\forall$:

Lemma 3.27. *Suppose $\langle T_i : i \in X \rangle$ combine to form S. Let r denote the root of S, and for each $i \in X$, let $r_{i,0}$ and $r_{i,1}$ denote the roots of the two copies of T_i in S (i.e., $r_{i,0} = r^\frown(i, 0)$ and $r_{i,1} = r^\frown(i, 1)$). Given any König cover (C, M) of S, for each $i \in X$, we can uniformly computably choose one of $r_{i,0}$ or $r_{i,1}$ (call our choice r_i) such that:*

- *$r_i \in S^*$;*
- *$r \notin C$ if and only if for all $i \in X$, $r_i \in C$.*

Therefore if $\langle T_n : n \in X \rangle$ codes the set $A \subseteq X$, then S codes the bit 0 if and only if $A = X$.

Proof. Given a König cover (C, M) of S and some $i \in X$, we choose r_i as follows. If neither $(r, r_{i,0})$ nor $(r, r_{i,1})$ lie in M, then define $r_i = r_{i,0} \in S^*$.

Otherwise, since M is a matching, exactly one of $(r, r_{i,0})$ and $(r, r_{i,1})$ lie in M, say $(r, r_{i,j})$. If $r \notin C$, we choose $r_i = r_{i,1-j} \in S^*$. If $r \in C$, note that since $(r, r_{i,j}) \in M$, we have (by duality) that $r_{i,j} \notin C$. Then we choose $r_i = r_{i,j} \in S^*$. This completes the definition of r_i.

If $r \notin C$, then for all $i \in X$ and $j < 2$, $r_{i,j} \in C$ because $(r, r_{i,j})$ must be covered by C. In particular, $r_i \in C$ for all $i \in X$.

If $r \in C$, then (by duality) there is a unique $i \in X$ and $j < 2$ such that $(r, r_{i,j}) \in M$. In that case, we chose $r_i = r_{i,j} \notin C$. $\square$

In the above lemma, it is important to note that our choice of each r_i depends on the König cover (C, M); in fact it depends on both C and M.

We can now use the combine operation to implement $\neg$.

Definition 3.28. The *complement* of T, denoted $\overline{T}$, is defined by combining the single-element sequence $\langle T \rangle$.

By Lemma 3.27, if T codes the bit i, then $\overline{T}$ codes the bit $1 - i$.

Next, we work towards iterating the combine operation to implement the jump, with the eventual goal of proving a generalization of [2, Lemma 4.7]. In order to reason about trees which are formed by iterating the combine operation, we generalize Lemma 3.27 slightly:

Lemma 3.29. *Suppose $\langle T_i : i \in X \rangle$ combine to form the subtree of S above some $r \in S$. For each $i \in X$, let $r_{i,0}$ and $r_{i,1}$ denote the roots of the two copies of T_i in*

S above r. Given any König cover (C, M) of S such that $r \in S^*$, for each i, we can uniformly computably choose one of $r_{i,0}$ or $r_{i,1}$ (call our choice r_i) such that

- *$r_i \in S^*$;*
- *$r \notin C$ if and only if for all $i \in X$, $r_i \in C$.*

Proof. By Lemma 3.24, (C, M) restricts to a König cover of the subtree of S above r. Apply Lemma 3.27 to the subtree of S above r, then use Proposition 3.25. □

We may now present a more general and more informative version of [2, Lemma 4.7].

Lemma 3.30. *Given a sequence of trees $\langle T_i : i \in \mathbb{N} \rangle$ (each with the empty node as root), we can uniformly compute a sequence of trees $\langle S_e : e \in \mathbb{N} \rangle$ (each with the empty node as root) such that given a König cover (C_e, M_e) of S_e, we can uniformly compute a sequence of sets of nodes $\langle R_{e,i} \rangle_i$ in S_e^* such that*

1. *each $r \in R_{e,i}$ has length two or three;*
2. *for each i and each $r \in R_{e,i}$, the subtree of S_e above r is $r^\frown T_i$;*
3. *if the set $A \subseteq \mathbb{N}$ is such that*

$$\begin{aligned} i \in A \quad &\Rightarrow \quad R_{e,i} \subseteq C_e \\ i \notin A \quad &\Rightarrow \quad R_{e,i} \subseteq \overline{C_e}, \end{aligned}$$

   then $e \in A'$ if and only if the root of S_e lies in C_e.

Therefore, if $\langle T_i \rangle$ codes a set A, then $\langle S_e \rangle$ codes A'.

Iterating the combine operation (as we will do in the following proof) introduces a complication, which necessitates the assumption in (3). For each e and i, instead of choosing a single node r_i as in Lemma 3.29, we now have to choose a set of nodes $R_{e,i}$. This is because we might want to copy the tree T_i more than twice, at multiple levels of the tree S_e. If T_i is not good (Definition 3.21), these copies could code different bits (according to appropriate restrictions of (C_e, M_e)), so we could have $R_{e,i} \not\subseteq C_e$ and $R_{e,i} \not\subseteq \overline{C_e}$. In that case, we have little control over whether the root of S_e lies in C_e.

Also, in the assumption of (3), we write $\Rightarrow$ instead of $\Leftrightarrow$ because writing $\Leftrightarrow$ would require us to specify separately that we do not restrict whether $i \in A$ in the case that $R_{e,i}$ is empty. (In the following proof, $R_{e,i}$ could be empty if the construction of S_e does not involve T_i at all.)

Proof of Lemma 3.30. We start by constructing S_e. Observe that $e \in A'$ if and only if

$$\neg\forall(\sigma, s) \in \{(\sigma, s) : \Phi^{\sigma}_{e,s}(e)\downarrow\}\neg\forall i \in \operatorname{dom}(\sigma)[(\sigma(i) = 1 \land i \in A) \\ \lor (\sigma(i) = 0 \land \neg(i \in A))].$$

Each occurrence of $\neg\forall$ or $\neg$ corresponds to one application of the combine operation in our construction of S_e.

Formally, for each finite partial $\sigma : \mathbb{N} \to 2$ and $i \in \operatorname{dom}(\sigma)$, define $T_i^{\sigma} = T_i$ if $\sigma(i) = 1$, otherwise define $T_i^{\sigma} = \overline{T_i}$. Now, for each σ and s such that $\Phi^{\sigma}_{e,s}(e)\downarrow$, define $T_{\sigma,s}$ by combining $\langle T_i^{\sigma} : i \in \operatorname{dom}(\sigma)\rangle$. Finally, combine $\langle T_{\sigma,s} : \Phi^{\sigma}_{e,s}(e)\downarrow\rangle$ to form S_e.

Next, given a König cover (C_e, M_e) of S_e, we construct $\langle R_{e,i}\rangle_i$ as follows. First apply Lemma 3.29 to $\langle T_{\sigma,s} : \Phi^{\sigma}_{e,s}(e)\downarrow\rangle$ and (C_e, M_e) to choose $\langle r_{\sigma,s} : \Phi^{\sigma}_{e,s}(e)\downarrow\rangle \subseteq S^*_e$ such that

- the subtree of S_e above each $r_{\sigma,s}$ is $r_{\sigma,s}{}^\frown T_{\sigma,s}$;
- the root of S_e lies in C_e if and only if there is some σ and s such that $\Phi^{\sigma}_{e,s}(e)\downarrow$ and $r_{\sigma,s} \notin C_e$.

Next, for each σ and s such that $\Phi^{\sigma}_{e,s}(e)\downarrow$, apply Lemma 3.29 to $\langle T^{\sigma}_i : i \in \mathrm{dom}(\sigma)\rangle$ and the König cover (C_e, M_e) restricted to the subtree of S_e above $r_{\sigma,s}$. This produces $\langle r^{\sigma,s}_i : i \in \mathrm{dom}(\sigma)\rangle \subseteq S^*_e$ (all extending $r_{\sigma,s}$) such that

- the subtree of S_e above each $r^{\sigma,s}_i$ is $r^{\sigma,s}_i{}^\frown T^{\sigma}_i$;
- $r_{\sigma,s} \notin C_e$ if and only if $r^{\sigma,s}_i \in C_e$ for all $i \in \mathrm{dom}(\sigma)$.

Finally, for each σ and s such that $\Phi^{\sigma}_{e,s}(e)\downarrow$ and each i such that $\sigma(i) = 0$, apply Lemma 3.29 to the single-element sequence $\langle T_i\rangle$ and (C_e, S_e) restricted to the subtree of S_e above $r^{\sigma,s}_i$ to obtain $\overline{r}^{\sigma,s}_i \in S^*_e$ extending $r^{\sigma,s}_i$ such that

- the subtree of S_e above $\overline{r}^{\sigma,s}_i$ is $\overline{r}^{\sigma,s}_i{}^\frown T_i$;
- $r^{\sigma,s}_i \in C_e$ if and only if $\overline{r}^{\sigma,s}_i \notin C_e$.

Define

$$R_{e,i} = \{r^{\sigma,s}_i : \Phi^{\sigma}_{e,s}(e)\downarrow, \sigma(i) = 1\} \cup \{\overline{r}^{\sigma,s}_i : \Phi^{\sigma}_{e,s}(e)\downarrow, \sigma(i) = 0\}.$$

First observe that each $r^{\sigma,s}_i$ has length two and each $\overline{r}^{\sigma,s}_i$ has length three. Hence (1) holds. Next, since $T^{\sigma}_i = T_i$ if $\sigma(i) = 1$, the subtree of S_e above each $r \in R_{e,i}$ is $r^\frown T_i$, i.e., (2) holds.

We prove that (3) holds. Suppose that $A \subseteq \mathbb{N}$ is such that

$$i \in A \quad \Rightarrow \quad R_{e,i} \subseteq C_e$$
$$i \notin A \quad \Rightarrow \quad R_{e,i} \subseteq \overline{C_e}.$$

Now, $e \in A'$ if and only if there is some $\sigma \prec A$ and s such that $\Phi^{\sigma}_{e,s}(e)\downarrow$. By our assumption on A and the definition of $R_{e,i}$, that holds if and only if there is some σ and s such that $\Phi^{\sigma}_{e,s}(e)\downarrow$ and for all $i \in \mathrm{dom}(\sigma)$:

$$\sigma(i) = 1 \quad \Leftrightarrow \quad r_i^{\sigma,s} \in C_e$$
$$\sigma(i) = 0 \quad \Leftrightarrow \quad \overline{r}_i^{\sigma,s} \notin C_e.$$

Chasing through the above definitions, we see that the above holds if and only if the root of S_e lies in C_e, as desired.

Finally, suppose that $\langle T_i \rangle$ codes the set A. We show that $\langle S_e \rangle$ codes A'. Fix a König cover $\langle (C_e, M_e) \rangle$ of $\langle S_e \rangle$. First we show that the assumption in (3) holds for A. Fix $e, i \in \mathbb{N}$. If $R_{e,i}$ is empty, the desired statement holds. Otherwise, fix $r \in R_{e,i}$. Since r lies in S_e^*, Lemma 3.24 says that (C_e, M_e) restricts to a König cover of the subtree of S_e above r. By (2), the subtree of S_e above r is $r^\frown T_i$. Since T_i codes $A(i)$, so does $r^\frown T_i$. We conclude that

$$r \in C_e \quad \Leftrightarrow \quad \text{the root of } T_i \in C_i \quad \Leftrightarrow \quad i \in A.$$

It follows that the assumption in (3) holds for A. Now by (3), $e \in A'$ if and only if the root of S_e lies in C_e.

Since this holds for every König cover $\langle (C_e, M_e) \rangle$ of $\langle S_e \rangle$, $\langle S_e \rangle$ codes A' as desired. □

Remark 3.31. In the proof of Lemma 3.30, we could just as well have defined $R_{e,i}$ to be the set of all nodes in S_e^* which are roots of copies of T_i. (Formally, for each $T_{\sigma,s}$ such that $\Phi^{\sigma}_{e,s}(e)\downarrow$, we could include the roots of the component T_i^{σ}'s if $\sigma(i) = 1$, and the roots of the component T_i's in the T_i^{σ}'s if $\sigma(i) = 0$, as long as they lie in S_e^*.)

Next, we make two small tweaks to Lemma 3.30. First, we adjust conclusion (3) to fit our definition of jump hierarchy (Definition 2.2). Second, we broaden the scope of our conclusions to include König covers of copies of S_n, not just König covers of S_n itself. Lemma 3.32 is the central lemma behind our reductions from ATR and ATR_2 to KDT.

Lemma 3.32. *Given a sequence of sequences of trees $\langle\langle T_n^a\rangle_n\rangle_a$ (each with the empty node as root), we can uniformly compute a sequence of trees $\langle S_n\rangle_n$ (each with the empty node as root) such that for any $s_n \in \mathbb{N}^{<\mathbb{N}}$ and any König cover (C_n, M_n) of $s_n{}^\frown S_n$, we can uniformly compute a sequence of sets of nodes $\langle R^a_{n,i}\rangle_{a,i}$ in $(s_n{}^\frown S_n)^*$ such that*

1. *each $r \in R^a_{n,i}$ has length two or three (plus the length of s_n);*
2. *for each a, i, and each $r \in R^a_{n,i}$, the subtree of $s_n{}^\frown S_n$ above r is $r^\frown T_i^a$;*
3. *suppose that for each a, the set $Y_a \subseteq \mathbb{N}$ is such that*

$$\begin{aligned} i \in Y_a \quad &\Rightarrow \quad R^a_{n,i} \subseteq C_n \\ i \notin Y_a \quad &\Rightarrow \quad R^a_{n,i} \subseteq \overline{C_n}, \end{aligned}$$

then $n \in (\bigoplus_a Y_a)'$ if and only if s_n lies in C_n.

Therefore, if for each a, $\langle T_n^a\rangle_n$ codes a set Y_a, then $\langle S_n\rangle_n$ codes $(\bigoplus_a Y_a)'$.

Proof. Apply Lemma 3.30 to $\langle T_n^a\rangle_{a,n}$. Given a König cover (C_n, M_n) of $s_n{}^\frown S_n$, we may compute the corresponding König cover of S_n (as we observed after Definition

3.21). Then apply Lemma 3.30 to obtain $\langle R^a_{n,i}\rangle_{n,i}$ in S^*_n. It is straightforward to check that $\langle s_n{}^\frown R^a_{n,i}\rangle_{n,i}$ satisfies conclusions (1)–(3). $\square$

As a warmup for our reduction from ATR_2 to KDT, we use Lemma 3.32 to prove that $\mathsf{ATR} \leq_W \mathsf{KDT}$. Our proof is essentially the same as that of [2, Theorem 4.11]. Note that we do not use the sets $R^a_{n,i}$ in the following proof, only the final conclusion of Lemma 3.32. (The sets $R^a_{n,i}$ will be used in our reduction from ATR_2 to KDT.)

Theorem 3.33. $\mathsf{ATR} \leq_W \mathsf{KDT}$.

Proof. We reduce the version of ATR in Proposition 2.6 to KDT. Given a labeled well-ordering $\mathcal{L}$ and a set A, we will use $(\mathcal{L}\oplus A)$-effective transfinite recursion on L to define an $(\mathcal{L}\oplus A)$-recursive function $f : L \to \omega$ such that for each $b \in L$, $\Phi^{\mathcal{L}\oplus A}_{f(b)}$ is interpreted as a sequence of trees $\langle T^b_n\rangle_n$ (each with the empty node as root). We will show that $\langle T^b_n\rangle_n$ codes the b^{th} column of the jump hierarchy on L which starts with A.

For the base case, we use Lemma 3.22 to compute a sequence of trees $\langle T^{0_L}_n\rangle_n$ which codes A. Otherwise, for $b >_L 0_L$, we use Lemma 3.32 to compute a sequence of trees $\langle T^b_n\rangle_n$ such that if for each $a <_L b$, $\Phi^{\mathcal{L}\oplus A}_{f(a)}$ is (interpreted as) a sequence of trees $\langle T^a_n\rangle_n$ which codes Y_a, then $\langle T^b_n\rangle_n$ codes $\left(\bigoplus_{a<_L b} Y_a\right)'$.

Note that f is total: for any b, we can interpret $\langle \Phi^{\mathcal{L}\oplus A}_{f(a)}\rangle_{a<_L b}$ as a sequence of sequences of trees and apply Lemma 3.32 to obtain $\langle T^b_n\rangle_n$. This also means that every $\langle T^b_n\rangle_n$ (for $b >_L 0_L$) was obtained using Lemma 3.32.

We may view the disjoint union of $\langle\langle T^b_n\rangle_n\rangle_{b\in L}$ as a KDT-instance. This defines the forward reduction from ATR to KDT.

For the backward reduction, let $\langle\langle(C_n^b, M_n^b)\rangle_n\rangle_{b\in L}$ be a solution to the above KDT-instance. We may uniformly decode said solution to obtain a sequence of sets $\langle Y_b\rangle_{b\in L}$.

By transfinite induction along L using Lemmas 3.22 and 3.32, $\langle T_n^b\rangle_n$ is good for all $b \in L$, and $\langle Y_b\rangle_{b\in L}$ is the jump hierarchy on L which starts with A. $\square$

What if we want to use the forward reduction from ATR to KDT in our reduction from ATR_2 to KDT? If the given ATR_2-instance $\mathcal{L}$ is ill-founded, things could go wrong in the "effective transfinite recursion". Specifically, there may be some $a \in L$ and $i \in \mathbb{N}$ such that T_i^a is not good, i.e., there may be some $r, s \in \mathbb{N}^{<\mathbb{N}}$ and some König covers of $r^\frown T_i^a$ and $s^\frown T_i^a$ which code different bits. In order to salvage the situation, we will modify the backward reduction to check for such inconsistencies. If they are present, we use them to compute an infinite $<_L$-descending sequence.

In order to detect inconsistencies, for each $b \in L$ and $n \in \mathbb{N}$, we need to keep track of the internal structure of (C_n^b, M_n^b) in the KDT-solution. According to Lemma 3.32 and our construction of T_n^b, for each $a <_L b$ and $i \in \mathbb{N}$, there is a set of nodes $R_{n,i}^a$ in $(T_n^b)^*$ such that:

- for each $r \in R_{n,i}^a$, the subtree of T_n^b above r is $r^\frown T_i^a$;
- if for each i, either $R_{n,i}^a \subseteq C_n^b$ or $R_{n,i}^a \subseteq \overline{C_n^b}$, then (C_n^b, M_n^b) codes the n^{th} bit of $(\bigoplus_a Y_a)'$, where each Y_a satisfies the assumption in Lemma 3.32(3).

The "consistent" case is if for each $a <_L b$ and $i \in \mathbb{N}$, (C_i^a, T_i^a) codes the same bit as the restriction of (C_n^b, M_n^b) to the subtree above each r in $R_{n,i}^a$. (This must happen if each T_i^a is good, but it could also happen "by chance".) We will show that this ensures that for each a and i, either $R_{n,i}^a \subseteq C_n^b$ or $R_{n,i}^a \subseteq \overline{C_n^b}$. Furthermore, for each a, the Y_a coded by $\langle T_i^a \rangle_i$ must satisfy the assumptions in Lemma 3.32(3), so we correctly calculate the next column of our jump hierarchy.

On the other hand, what if there are some $a <_L b$, $i \in \mathbb{N}$, and $r_0 \in R_{n,i}^a$ such that (C_i^a, M_i^a) codes a different bit from the restriction of (C_n^b, M_n^b) to the subtree above r_0? Then consider T_i^a and the subtree of T_n^b above r_0. The latter tree is a copy of T_i^a (specifically, it is $r_0 {}^\frown T_i^a$), yet its König cover codes a different bit from that of T_i^a, so we can use Lemma 3.32 to find a subtree of T_i^a and a subtree of T_n^b above r_0 (both subtrees are copies of $T_{i_0}^{a_0}$ for some $a_0 <_L a$, $i_0 \in \mathbb{N}$) on which appropriate restrictions of (C_i^a, M_i^a) and (C_n^b, M_n^b) code different bits. By repeating this process, we can obtain an infinite $<_L$-descending sequence.

In order to formalize the above arguments, we organize the above recursive process using the sets $R_{n,i}^{b,a}$, defined as follows:

Definition 3.34. Fix a labeled linear ordering $\mathcal{L}$ and use the forward reduction in Theorem 3.33 to compute $\langle \langle T_n^b \rangle_n \rangle_{b \in L}$. For each n and b, fix a König cover (C_n^b, M_n^b) of T_n^b. For each $a <_L b$ and each $i, n \in \mathbb{N}$, we define a set of nodes $R_{n,i}^{b,a}$ in T_n^b as follows: $R_{n,i}^{b,a}$ is the set of all r for which there exist $j \geq 1$ and

$$\begin{array}{ccccccccc} \langle\rangle = r_0 & \prec & r_1 & \prec & \cdots & \prec & r_j = r & \text{in } T_n^b \\ b = c_0 & >_L & c_1 & >_L & \cdots & >_L & c_j = a & \text{in } L \\ n = i_0 & , & i_1 & , & \cdots & , & i_j = i & \text{in } \mathbb{N} \end{array}$$

such that for all $0 < l \leq j$, r_l lies in $R_{i_{l-1},i_l}^{c_l}$ as calculated by applying Lemma 3.32 to (C_n^b, M_n^b) restricted to the subtree of T_n^b above r_{l-1}.

We make two easy observations about $R_{n,i}^{b,a}$:

1. By induction on l, r_l lies in $(T_n^b)^*$ and the subtree of T_n^b above r_l is $r_l {}^\frown T_{i_l}^{c_l}$.

In particular, for each $r \in R^{b,a}_{n,i}$, $r \in (T^b_n)^*$ and the subtree of T^b_n above r is $r^\frown T^a_i$.

2. $R^{b,a}_{n,i}$ is uniformly c.e. in $\mathcal{L} \oplus (C^b_n, M^b_n)$. (A detailed analysis shows that $R^{b,a}_{n,i}$ is uniformly computable in $\mathcal{L} \oplus (C^b_n, M^b_n)$, but we do not need that.)

With the $R^{b,a}_{n,i}$'s in hand, we can make precise what we mean by consistency:

Definition 3.35. In the same context as the previous definition, we say that $a \in L$ is *consistent* if for all $i \in \mathbb{N}$:

$$\begin{aligned} \text{the root of } T^a_i \in C^a_i \quad &\Rightarrow \quad R^{b,a}_{n,i} \subseteq C^b_n \text{ for all } b >_L a, n \in \mathbb{N} \\ \text{the root of } T^a_i \notin C^a_i \quad &\Rightarrow \quad R^{b,a}_{n,i} \subseteq \overline{C^b_n} \text{ for all } b >_L a, n \in \mathbb{N}. \end{aligned}$$

Observe that if T^a_i is good for all i, then observation (1) above implies that a is consistent, regardless of what $\langle (C^b_n, M^b_n) \rangle_{b,n}$ may be. However, unless L is well-founded, we cannot be certain that T^a_i is good. Consistency is a weaker condition which suffices to ensure that we can still obtain a jump hierarchy on L, as we show in Corollary 3.38. We will also show that inconsistency cannot come from nowhere, i.e., if b_0 is inconsistent, then there is some $b_1 <_L b_0$ which is inconsistent, and so on, yielding an infinite $<_L$-descending sequence of inconsistent elements.

Furthermore, consistency is easy to check: by observation (2) above, whether a is consistent is Π^0_1 (in $\mathcal{L} \oplus \langle (C^b_n, M^b_n) \rangle_{b,n}$).

We prove two lemmas that will yield the desired result when combined:

Lemma 3.36. *Fix König covers $\langle (C^b_n, M^b_n) \rangle_{b,n}$ for $\langle T^b_n \rangle_{b,n}$. Now fix n and b. Suppose that for each $a <_L b$, the set $Y_a \subseteq \mathbb{N}$ is such that*

$$\begin{aligned} i \in Y_a \quad &\Rightarrow \quad R^{b,a}_{n,i} \subseteq C^b_n \\ i \notin Y_a \quad &\Rightarrow \quad R^{b,a}_{n,i} \subseteq \overline{C^b_n}. \end{aligned}$$

Then $n \in \left(\bigoplus_{a<_L b} Y_a\right)'$ *if and only if the root of* T^b_n *lies in* C^b_n.

Proof. Recall that $\langle T^b_n\rangle_{n\in\mathbb{N}}$ is computed by applying Lemma 3.32 to $\langle\langle T^a_n\rangle_{n\in\mathbb{N}}\rangle_{a<_L b}$. By definition of $R^{b,a}_{n,i}$, $R^a_{n,i}$ (as obtained from Lemma 3.32) is a subset of $R^{b,a}_{n,i}$ (this is the case $j = 1$). So for all $a <_L b$,

$$\begin{aligned} i \in Y_a \quad &\Rightarrow \quad R^a_{n,i} \subseteq R^{b,a}_{n,i} \subseteq C^b_n \\ i \notin Y_a \quad &\Rightarrow \quad R^a_{n,i} \subseteq R^{b,a}_{n,i} \subseteq \overline{C^b_n}. \end{aligned}$$

The desired result follows from Lemma 3.32(3). □

Lemma 3.37. *Fix König covers* $\langle (C^c_m, M^c_m)\rangle_{c,m}$ *for* $\langle T^c_m\rangle_{c,m}$*. Now fix* m *and* $b <_L c$*. Suppose that for each* $a <_L b$*, the set* $Y_a \subseteq \mathbb{N}$ *is such that*

$$\begin{aligned} i \in Y_a \quad &\Rightarrow \quad R^{c,a}_{m,i} \subseteq C^c_m \\ i \notin Y_a \quad &\Rightarrow \quad R^{c,a}_{m,i} \subseteq \overline{C^c_m}. \end{aligned}$$

Then for all $n \in \mathbb{N}$*,*

$$\begin{aligned} n \in \left(\bigoplus_{a<_L b} Y_a\right)' \quad &\Rightarrow \quad R^{c,b}_{m,n} \subseteq C^c_m \\ n \notin \left(\bigoplus_{a<_L b} Y_a\right)' \quad &\Rightarrow \quad R^{c,b}_{m,n} \subseteq \overline{C^c_m}. \end{aligned}$$

Proof. If $R^{c,b}_{m,n}$ is empty, then the desired result is vacuously true. Otherwise, consider $r \in R^{c,b}_{m,n}$. As we observed right after Definition 3.34, $r \in (T^c_m)^*$ and the subtree of T^c_m above r is $r^\frown T^b_n$. T^b_n was constructed by applying Lemma 3.32 to $\langle\langle T^a_n\rangle_{n\in\mathbb{N}}\rangle_{a<_L b}$, so we can use the restriction of (C^c_m, M^c_m) to $r^\frown T^b_n$ to compute sets $\langle R^a_{n,i}\rangle_{a<_L b, i\in\mathbb{N}}$ of nodes in $(r^\frown T^b_n)^*$ satisfying the conclusions of Lemma 3.32.

We claim that for all $a <_L b$, $R^a_{n,i} \subseteq R^{c,a}_{m,i}$.

Proof of claim. Consider $s \in R^a_{n,i}$. We know that s extends r and $r \in R^{c,b}_{m,n}$. Fix $j \geq 1$ and

$$\begin{array}{ccccccccc} \langle\rangle = r_0 & \prec & r_1 & \prec & \cdots & \prec & r_j = r & \text{in } T^c_m \\ c = c_0 & >_L & c_1 & >_L & \cdots & >_L & c_j = b & \text{in } L \\ m = i_0 & , & i_1 & , & \cdots & , & i_j = n & \text{in } \mathbb{N} \end{array}$$

which witness that $r \in R^{c,b}_{m,n}$. Then we can append one column:

$$\begin{array}{ccccccccccc} \langle\rangle = r_0 & \prec & r_1 & \prec & \cdots & \prec & r_j = r & \prec & r_{j+1} = s & \text{in } T^c_m \\ c = c_0 & >_L & c_1 & >_L & \cdots & >_L & c_j = b & >_L & c_{j+1} = a & \text{in } L \\ m = i_0 & , & i_1 & , & \cdots & , & i_j = n & , & i_{j+1} = i & \text{in } \mathbb{N} \end{array}$$

Since $s \in R^a_{n,i}$, this witnesses that $s \in R^{c,a}_{m,i}$. □

By our claim, we have that

$$\begin{aligned} i \in Y_a \quad &\Rightarrow \quad R^a_{n,i} \subseteq R^{c,a}_{m,i} \subseteq C^c_m \\ i \notin Y_a \quad &\Rightarrow \quad R^a_{n,i} \subseteq R^{c,a}_{m,i} \subseteq \overline{C^c_m}. \end{aligned}$$

By Lemma 3.32(3), $n \in \left(\bigoplus_{a<_L b} Y_a\right)'$ if and only if $r \in C^c_m$. This concludes the proof. □

Putting the previous two lemmas together, we obtain

Corollary 3.38. *Fix König covers $\langle (C^b_n, M^b_n)\rangle_{b,n}$ for $\langle T^b_n\rangle_{b,n}$. For each $b \in L$, define Y_b by decoding $\langle (C^b_n, M^b_n)\rangle_n$, i.e.,*

$$Y_b = \{n \in \mathbb{N} : \textit{the root of } T^b_n \textit{ lies in } C^b_n\}.$$

If all $a <_L b$ are consistent, then b is consistent and $Y_b = \left(\bigoplus_{a<_L b} Y_a\right)'$.

Proof. 0_L is consistent because every $T_n^{0_L}$ is good (Lemma 3.22). Consider now any $b >_L 0_L$. Every $a <_L b$ is consistent, so for all $a <_L b$:

$$i \in Y_a \quad \Rightarrow \quad R_{m,i}^{c,a} \subseteq C_m^c \text{ for all } c >_L a, m \in \mathbb{N}$$
$$i \notin Y_a \quad \Rightarrow \quad R_{m,i}^{c,a} \subseteq \overline{C_m^c} \text{ for all } c >_L a, m \in \mathbb{N}.$$

By Lemma 3.36, $Y_b = \left(\bigoplus_{a<_L b} Y_a\right)'$.

Also, by Lemma 3.37, for all $n \in \mathbb{N}$:

$$n \in \left(\bigoplus_{a<_L b} Y_a\right)' \quad \Rightarrow \quad R_{m,n}^{c,b} \subseteq C_m^c \text{ for all } c >_L b, m \in \mathbb{N}$$
$$n \notin \left(\bigoplus_{a<_L b} Y_a\right)' \quad \Rightarrow \quad R_{m,n}^{c,b} \subseteq \overline{C_m^c} \text{ for all } c >_L b, m \in \mathbb{N}.$$

It follows that b is consistent. □

We are finally ready to construct a reduction from ATR_2 to KDT.

Theorem 3.39. $\mathsf{ATR}_2 \leq_W \mathsf{LPO} * \mathsf{KDT}$. *In particular,* $\mathsf{ATR}_2 \leq_c \mathsf{KDT}$ *and* $\mathsf{ATR}_2 \leq_W^{\text{arith}} \mathsf{KDT}$.

Proof. Given a labeled linear ordering $\mathcal{L}$ (we may assume that L is labeled by Proposition 3.12) and a set A, we apply the forward reduction in Theorem 3.33 to produce some KDT-instance $\langle T_n^b \rangle_{b,n}$. For the backward reduction, given a KDT-solution $\langle\langle (C_n^b, M_n^b)\rangle_n\rangle_{b\in L}$, we start by uniformly decoding it to obtain a sequence of sets $\langle Y_b \rangle_{b\in L}$.

Next, since $R_{n,i}^{b,a}$ is uniformly c.e. in $\mathcal{L} \oplus (C_n^b, M_n^b)$, whether some $a \in L$ is inconsistent is uniformly c.e. in $\mathcal{L} \oplus \langle (C_n^b, M_n^b)\rangle_{b,n}$. Therefore we can use LPO (Definition 1.10) to determine whether every $a \in L$ is consistent.

If so, by Corollary 3.38, $\langle Y_b \rangle_{b\in L}$ is a jump hierarchy on L which starts with A.

If not, by Corollary 3.38, every inconsistent element is preceded by some other inconsistent element. Since whether some $a \in L$ is inconsistent is uniformly c.e. in $\mathcal{L} \oplus \langle (C_n^b, M_n^b) \rangle_{b,n}$, we can use it to compute an infinite $<_L$-descending sequence of inconsistent elements. □

3.2.2 Reducing KDT to ATR_2

This section presumes an understanding of the proofs in Simpson [41]. First, he proved in ATR_0 that for any set G, there is a countable coded ω-model of Σ^1_1-AC which contains G. His proof [41, Lemma 1] also shows that

Lemma 3.40. *If $\langle X_a \rangle_{a \in L}$ is a jump hierarchy on L and I is a proper cut of L which is not computable in $\langle X_a \rangle_{a \in L}$, then the countable coded ω-model $\mathcal{M} = \{A : \exists a \in I (A \leq_T X_a)\}$ satisfies Σ^1_1-AC.*

Sketch of proof. Given an instance $\varphi(n, Y)$ of Σ^1_1-AC, for each n, let $a_n \in I$ be $<_L$-least such that X_{a_n} computes a solution to $\varphi(n, \cdot)$. Since I is a proper cut, for any $a \in I$ and $b \in L \backslash I$, X_b computes every X_a-hyperarithmetic set. Therefore if $b \in L \backslash I$, then X_b computes $(a_n)_{n \in \omega}$.

Hence $(a_n)_{n \in \omega}$ is not cofinal in I, otherwise I would be computable in $\langle X_a \rangle_{a \in L}$. Fix $b \in I$ which bounds $(a_n)_{n \in \omega}$. Then there is a Σ^1_1-AC-solution to φ which is arithmetic in X_b (and hence lies in $\mathcal{M}$), as desired. □

We now adapt [41]'s proof of König's duality theorem in ATR_0 to show that

Theorem 3.41. KDT *is arithmetically Weihrauch reducible to* ATR_2.

Proof. Given a bipartite graph G, we would like to use ATR_2 to produce a countable coded ω-model of Σ^1_1-AC which contains G. In order to do that, we define a G-computable linear ordering (i.e., an instance of ATR_2) using the recursion theorem, as follows.

First define a predicate $P(G, e, X)$ to hold if X is a jump hierarchy on L^G_e which starts with G and does not compute any proper cut in L^G_e. Notice that $P(G, e, X)$ is arithmetic.

The total G-computable function to which we apply the recursion theorem is as follows. Given any G-computable linear ordering L^G_e, use Lemma 1.18 to define the G-computable tree H^G_e whose paths (if any) are solutions to $P(G, e, \cdot)$ (with Skolem functions). Then output an index for the Kleene-Brouwer ordering of H^G_e.

By the recursion theorem, we can G-uniformly compute a fixed point e for the above computable transformation. Observe that the following are (consecutively) equivalent:

1. L^G_e has an infinite G-hyperarithmetic descending sequence;
2. H^G_e has a G-hyperarithmetic path;
3. $P(G, e, \cdot)$ has a G-hyperarithmetic solution, i.e., there is a G-hyperarithmetic jump hierarchy on L^G_e which starts with G and does not compute any proper cut in L^G_e;
4. L^G_e is well-founded.

(The only nontrivial implication is (3) $\Rightarrow$ (4), which holds because no jump hierarchy on a G-computable ill-founded linear ordering can be G-hyperarithmetic; see Sacks [39, III.3.3].) But (1) and (4) contradict each other, so (1)–(4) are all false.

Hence L_e^G must be ill-founded and cannot have any infinite G-hyperarithmetic descending sequence. It follows that every infinite L_e^G-descending sequence defines a proper cut in L_e^G.

Next, we show that given an ATR_2-solution to L_e^G, we can arithmetically uniformly compute some proper cut I in L_e^G and a solution to $P(G, e, \cdot)$, i.e., a jump hierarchy $\langle X_a \rangle_{a \in L_e^G}$ which does not compute any proper cut in L_e^G. Then by Lemma 3.40, the countable coded ω-model of all sets which are computable in some X_a, $a \in I$, satisfies Σ_1^1-AC as desired.

If ATR_2 gives us an infinite L_e^G-descending sequence S, then we can use S to arithmetically uniformly compute a proper cut in L_e^G. Since L_e^G is the Kleene-Brouwer ordering of H_e^G, we can also use S to arithmetically uniformly compute a path on H_e^G. From said path, we can uniformly compute a solution to $P(G, e, \cdot)$.

If ATR_2 gives us a jump hierarchy X on L_e^G, we show how to arithmetically uniformly compute an infinite L_e^G-descending sequence. We may then proceed as in the previous case.

First arithmetically uniformly check whether X computes any proper cut in L_e^G. If so, we can arithmetically uniformly find an index for such a computation, and produce a proper cut in L_e^G. From that, we may uniformly compute an infinite L_e^G-descending sequence. If not, then X is a solution to $P(G, e, \cdot)$, so we can arithmetically uniformly compute a path on H_e^G, and hence an infinite L_e^G-descending sequence.

We have produced a countable coded ω-model of Σ_1^1-AC which contains the given graph G. Call it $\mathcal{M}$.

With $\mathcal{M}$ in hand, we follow the rest of Simpson's [41] proof in order to obtain a KDT-solution to G. His idea is to "relativize" Aharoni, Magidor, Shore's [2] proof of KDT in Π^1_1-CA_0 to $\mathcal{M}$. In the following, we will often write $\mathcal{M}$ instead of "the code of $\mathcal{M}$".

Let $G = (X, Y, E)$. (If we are not given a partition (X, Y) of the vertex set of G witnessing that G is bipartite, we can arithmetically uniformly compute such a partition.) If $A \subseteq X$, then the *demand set of* A is defined by

$$D_G(A) = \{y \in Y : xEy \to x \in A\}.$$

Note that if $A \in \mathcal{M}$, then $D_G(A)$ is uniformly arithmetic in $\mathcal{M}$ and the code of A.

Next, consider the set of pairs

$$S = \{(A, F) \in \mathcal{M} : A \subseteq X \text{ and } F : A \to D_G(A) \text{ is a matching}\}.$$

(Note that A and F may be infinite.) S (specifically the set of codes of $(A, F) \in S$) is arithmetic over $\mathcal{M}$. So is the set $\bigcup\{A : (A, F) \in S\} \subseteq X$, which we denote by A^*.

Next, for each $x \in A^*$, we define $F^*(x)$ to be $F(x)$, where (A, F) is the least (with respect to the enumeration of $\mathcal{M}$) pair in S such that $x \in A$. Then $F^* : A^* \to D_G(A^*)$ is a matching ([41, Lemma 2]). Note that F^* is arithmetic over $\mathcal{M}$.

Next, define $X^* = X - A^*$ and $Y^* = Y - D_G(A^*)$. Both sets are arithmetic over $\mathcal{M}$. Simpson then constructs (by recursion along ω) a matching H from Y^* to X^* which is arithmetic in $G \oplus \mathcal{M}$, as follows. Each step of the recursion proceeds by searching for a pair of adjacent vertices (one in X^*, one in Y^*) whose removal does not destroy *goodness*: a cofinite induced subgraph G' (with vertices partitioned into $X' \subseteq X$ and $Y' \subseteq Y$) of G is *good* if for any $A \subseteq X'$ in $\mathcal{M}$ and any matching

$F : A \to D_{G'}(A)$ in $\mathcal{M}$, $D_{G'}(A) - \text{range}(F)$ and Y^* are disjoint. (This definition is not related to Definition 3.21.) This recursion eventually matches every vertex in Y^* to some vertex in X^* ([41, Lemmas 3, 5]).

The property of goodness (where each G' is encoded by the finite set of vertices in $G \backslash G'$) is arithmetic over $\mathcal{M}$. Hence the resulting matching H is arithmetic over $\mathcal{M}$.

Finally, we arrive at a KDT-solution to G: $F^* \cup H$ is a matching in G, with corresponding dual cover $A^* \cup Y^*$. $(F^* \cup H, A^* \cup Y^*)$ can be arithmetically uniformly computed from $\mathcal{M}$. □

Using Theorems 3.39 and 3.41, we conclude that

Corollary 3.42. ATR_2 *and* KDT *are arithmetically Weihrauch equivalent.*

CHAPTER 4

DIFFERENT WAYS OF COMPOSING MULTIVALUED FUNCTIONS

In this chapter, we compare and contrast different methods of composing multivalued functions. As a motivating example, consider Ramsey's theorem for k-colorings of n-tuples (RT^n_k): for every coloring $c : [\mathbb{N}]^n \to k$, there is an infinite c-homogeneous set. Then $\mathsf{RCA}_0 + \mathsf{RT}^n_3 \vdash \mathsf{RT}^n_2$ (view the given 2-coloring as a 3-coloring). This proof only invokes RT^n_3 once, and it can be translated into a Weihrauch reduction from RT^n_2 to RT^n_3.

Less trivially, we also have that $\mathsf{RCA}_0 + \mathsf{RT}^n_2 \vdash \mathsf{RT}^n_3$. The usual proof invokes RT^n_2 twice, in series: given a 3-coloring of $[\mathbb{N}]^n$ by red, green, and blue, first define a 2-coloring of $[\mathbb{N}]^n$ by red and "grue". Then use RT^n_2 to obtain an infinite homogeneous set for it. If we obtain a red homogeneous set, then we are done. If we obtain a "grue" homogeneous set, then we apply RT^n_2 to the original coloring restricted to this set, and we are done.

Is there a proof of RT^n_3 which only invokes RT^n_2 once?[1] If not, is there a proof of RT^n_3 which invokes RT^n_2 twice, but in parallel?[2] We want to study such questions from the point of view of Weihrauch reducibility. In order to do so, we must define some reducibility which would capture the notion of P being reducible to multiple instances of Q in series. There are three known ways to formalize this idea:

1. the compositional product (Definition 4.4);
2. reduction games (Definition 4.9);

[1] In the reverse mathematics setting, Hirst and Mummert [25] gave such a proof in RCA_0. Their proof was not "uniform". In the setting of Weihrauch reducibility, Hirschfeldt and Jockusch [24], Brattka and Rakotoniaina [10], and Patey [34] independently showed that there is no reduction.

[2] Note that invoking a theorem in parallel is a special case of invoking a theorem in series.

3. the step product (Definition 4.17).

In this chapter, we clarify the relationships between these three notions (for example, Theorems 4.22, 4.26, Corollary 4.28). We conclude that they are (mostly) equivalent, and hence one is (mostly) free to use whichever definition is convenient for one's purposes. Along the way, we prove some basic properties of these notions, and give counterexamples where appropriate.

We are also interested in capturing the notion of P being reducible to different theorems $Q_0, \dots, Q_{n-1}$ in series. One motivating example is Cholak, Jockusch, and Slaman's [12] proof of RT^2_2 which proceeds by first using one theorem to obtain an infinite set on which the given coloring is stable, and then restricting to said set and obtaining, by another theorem, an infinite homogeneous set. To formalize this notion, we consider a generalized reduction game and show how it relates to the other formalizations (Theorem 4.33).

Next, we give some notation and basic definitions. In this chapter, we only consider multivalued functions from $\mathbb{N}^\mathbb{N}$ to $\mathbb{N}^\mathbb{N}$, rather than multivalued functions between represented spaces in general. (We will argue that for our purposes, this is without loss of generality.) If Φ is a Turing functional and X is an oracle for Φ, we will sometimes write $\Phi(X)$ instead of Φ^X. Since Φ formally only takes numbers as input, this should not cause confusion.

A useful notion is that of a uniformly computable multivalued function: a multivalued function P is *uniformly computable* if it has a computable realizer; that is, there is a Turing functional Γ such that for every P-instance X, $\Gamma(X)$ is a P-solution to X. Note that the uniformly computable multivalued functions do not all lie in the same Weihrauch degree.[3]

[3]In fact, it is easy to see that the Medvedev degrees embed into the set of Weihrauch degrees

4.1 Formalizing compositions

In this section, we present several ways to formalize what it means for P to be reducible to multiple instances of Q, and prove some basic properties about them. Some of these definitions have been stated in chapter 1, but we repeat them for the reader's convenience.

4.1.1 Parallel product

We begin by considering what it means for P to be reducible to multiple instances of Q in parallel. This notion is captured by the parallel product:

Definition 4.1 (Brattka, Gherardi [6]). Given multivalued functions P and Q, the *parallel product* $P \times Q$ is the Cartesian product of P and Q. That is, instances are pairs (X, Y), where X is a P-instance and Y is a Q-instance. (Z, W) is a $(P \times Q)$-solution to (X, Y) if Z is a P-solution to X and W is a Q-solution to Y.

For example, we have that $\mathsf{RT}^n_j \times \mathsf{RT}^n_k \leq_W \mathsf{RT}^n_{jk}$: given a j-coloring and a k-coloring, we can pair them to obtain a jk-coloring. A homogeneous set for the jk-coloring will be homogeneous for both the j-coloring and the k-coloring. For other examples, see [7] and [15].

Up to Weihrauch degree, the parallel product is well-defined, associative, and monotone in both components [6, Proposition 3.2].

While we will not study the parallel product in this chapter, we will use it to state a later definition.

which contain a uniformly computable multivalued function. See [8, Theorem 9.1].

4.1.2 Compositional product

In this section, we define the compositional product of multivalued functions (Brattka, Gherardi, Marcone [7]; Brattka, Pauly [9]), which attempts to capture the notion of P being reducible to multiple instances of Q in series. We begin by defining the composition of multivalued functions, which forms a building block for the compositional product. Intuitively, $Q \circ P$ corresponds to invoking P and then Q, with no extra steps allowed in between; that is, the solution to the P-instance has to be a Q-instance.

Definition 4.2. Given multivalued functions P and Q, their *composition* $Q \circ P$ is the following multivalued function. Instances are P-instances X such that every P-solution Y to X is itself a Q-instance. Z is a $(Q \circ P)$-solution to X if there is some P-solution Y to X such that Z is a Q-solution to Y.

Note that the composition of P and Q as multivalued functions is more restrictive than the composition of P and Q as relations. This restriction implies that, for example, the composition of realizers for P and Q is a realizer of the composition $Q \circ P$.

It is easy to see that $\circ$ is associative:

Proposition 4.3. $\circ$ *is associative up to equality of multivalued functions; that is, for multivalued functions* P, Q, R, *we have* $(R \circ Q) \circ P = R \circ (Q \circ P)$.

However, $\circ$ is not monotone (in either component) with respect to Weihrauch reducibility. To illustrate what can go wrong, here are some examples.

1. Take any Q which is not uniformly computable and has a computable instance X_0 with a computable solution. (For example, take Q to be RT^1_2.)

Take P_0 to be the identity function, and take P_1 to be the identity function restricted to $\{X_0\}$. It is easy to see that $P_0 \leq_W P_1$ and $Q \circ P_0 \not\leq_W Q \circ P_1$.

2. Take any P which is not uniformly computable. For $i = 0, 1$, define P_i as follows: P_i-instances are P-instances, and (i, Y) is a P_i-solution to X if and only if Y is a P-solution to X. Define Q as follows: instances are pairs (i, Y), for any set Y and $i = 0, 1$. For each $(0, Y)$, Y is the only Q-solution, and for each $(1, Y)$, 0 is the only Q-solution. It is easy to see that $P_0 \leq_W P_1$ and $Q \circ P_0 \not\leq_W Q \circ P_1$.

3. Take any R which is not uniformly computable. Define P as follows: instances are pairs (i, X) for any set X and $i = 0, 1$. For each $(0, X)$, the P-solutions are pairs $(0, Y)$, where Y is an R-solution to X, and for each $(1, X)$, 0 is the only P-solution. Define Q_0 to be the identity function restricted to instances $(0, Y)$, for any set Y. Define Q_1 to be the identity function with only one instance 0. It is easy to see that $Q_0 \leq_W Q_1$ and $Q_0 \circ P \not\leq_W Q_1 \circ P$.

4. Define P as follows: instances are pairs (i, X) for any set X and $i = 0, 1$. Each (i, X) has a unique P-solution $(0, X)$. For $i = 0, 1$, define Q_i to be the identity function restricted to pairs (i, X). We have that $Q_0 \leq_W Q_1$. But $Q_0 \circ P$ has nonempty domain while $Q_1 \circ P$ has empty domain, so $Q_0 \circ P \not\leq_W Q_1 \circ P$.

Having defined $\circ$, we are now ready to define the compositional product $Q * P$, which attempts to capture the power of one invocation of P, followed by one invocation of Q in series.

Definition 4.4 (Brattka, Gherardi, Marcone [7]; Brattka, Pauly [9])**.** The *compositional product*[4] of Weihrauch degrees $\mathbf{p}$ and $\mathbf{q}$, written $\mathbf{q} * \mathbf{p}$, is defined to be the

[4]Brattka and Pauly [9] give a different definition of $\mathbf{q} * \mathbf{p}$ and show that it is equal to the

Weihrauch degree $\sup\{Q \circ P : Q \leq_W \mathbf{q},\ P \leq_W \mathbf{p}\}$.

That the supremum in the definition exists is in fact a theorem:

Theorem 4.5 (Brattka, Pauly [9, Corollaries 18, 20])**.** *For every* $\mathbf{p}$ *and* $\mathbf{q}$*, there are multivalued functions* P *of degree* $\mathbf{p}$ *and* Q *of degree* $\mathbf{q}$ *such that* $Q \circ P$ *has degree* $\mathbf{q} * \mathbf{p}$*.*

We abuse notation and use $Q * P$ to refer to the Weihrauch degree $\mathbf{q} * \mathbf{p}$, where P has degree $\mathbf{p}$ and Q has degree $\mathbf{q}$. Since $*$ is monotone in both coordinates, this is well-defined.

In order to state more facts about the compositional product, we use the notion of a cylinder due to Brattka and Gherardi [6]. We say that a multivalued function P is a *cylinder* if $P \equiv_{sW} \mathrm{id} \times P$. It is easy to see that if $Q \leq_W P$, then $Q \leq_{sW} \mathrm{id} \times P$. Therefore, if P is a cylinder, then $Q \leq_W P$ if and only if $Q \leq_{sW} P$.

The compositional product has a so-called cylindrical decomposition:

Lemma 4.6 (Brattka, Pauly [9, Lemma 21])**.** *For all* P *and* Q *which are cylinders, there exists a computable function* K *such that* $Q * P \equiv_W Q \circ K \circ P$*. Furthermore,* $Q \circ K \circ P$ *is a cylinder.*

We also have that

Proposition 4.7 (Brattka, Pauly [9, Proposition 32])**.** $*$ *is associative.* $*$ *is monotone in both components with respect to Weihrauch reducibility.*

supremum of all $Q \circ P$, where $Q \leq_W \mathbf{q}$ and $P \leq_W \mathbf{p}$ are multivalued functions on arbitrary represented spaces, not just $\mathbb{N}^{\mathbb{N}}$. Nevertheless, this definition is equivalent to theirs: suppose f is a multivalued function from (X, δ_X) to (Y, δ_Y) and g is a multivalued function from (Y, δ_Y) to (Z, δ_Z). Then $f \equiv_W \delta_Y \circ f \circ \delta_X^{-1}$, $g \equiv_W \delta_Z \circ g \circ \delta_Y^{-1}$, and $g \circ f \equiv_W (\delta_Z \circ g \circ \delta_Y^{-1}) \circ (\delta_Y \circ f \circ \delta_X^{-1})$.

In order to prove our main results, we will use the following version of Theorem 4.5 for multiple multivalued functions.

Lemma 4.8. *For every $Q_0, \ldots, Q_{n-1}$, there are multivalued functions $R_0, \ldots, R_{n-1}$ such that for each $i < n$, $R_i \leq_W Q_i$, and $Q_{n-1} * \cdots * Q_0 \equiv_W R_{n-1} \circ \cdots \circ R_0$.*

Proof. First, by replacing each Q_i with $\mathrm{id} \times Q_i$, we may assume that each Q_i is a cylinder. Next, by induction using Lemma 4.6, we obtain computable functions $K_0, \ldots, K_{n-2}$ such that

$$Q_{n-1} * \cdots * Q_0 \equiv_W Q_{n-1} \circ K_{n-2} \circ Q_{n-2} \circ \cdots \circ K_0 \circ Q_0.$$

Then define $R_{n-1} = Q_{n-1}$, and for $i < n-1$, define $R_i = K_i \circ Q_i$. For each i, it is easy to see that $R_i \leq_W Q_i$. □

4.1.3 Reduction games

In this section, we present another formalization of the notion of P being reducible to multiple instances of Q in series. The process of solving an instance of P using multiple instances of Q in series can be thought of as a game. Roughly speaking, Player I starts by posing a P-instance for Player II to solve. At each turn, Player II has oracle access to all of Player I's previous plays, and it can either compute a Q-instance for Player I to solve, or it can win by computing a solution to the P-instance posed by Player I.

Definition 4.9 (Hirschfeldt, Jockusch [24, Definition 4.1])**.** Define the *game reducing P to Q* as follows. In round $n = 1$, Player I starts by playing a P-instance X_0. Player II responds with either of the following:

– an X_0-computable Q-instance Y_1;

– an X_0-computable P-solution to X_0;

and an indication of which case it is (for the second case, Player II declares victory.)

In round $n > 1$, Player I plays a solution X_{n-1} to the Q-instance Y_{n-1}. Player II responds with either of the following:

– a $\left(\bigoplus_{i<n} X_i\right)$-computable Q-instance Y_n;

– a $\left(\bigoplus_{i<n} X_i\right)$-computable P-solution to X_0;

and an indication of which case it is (for the second case, Player II declares victory.)

Player II wins if it ever declares victory, after which the game ends. Otherwise Player I wins, which happens either if the game goes on forever, or Player II cannot move (which can only happen in the first round).

In the game reducing P to Q, even though II can only play sets which are computable in the join of all of I's previous plays, II is allowed to employ non-uniform strategies to decide which set to play. Since we are interested in solving P uniformly from multiple instances of Q, we will only consider computable strategies for II, defined as follows.

First we define some notation. If Z is a set and Φ is a Turing functional, then we define $\widehat{\Phi}^Z$ to be $\{n : \Phi^Z(n+1)\downarrow = 1\}$. Also, following [24], we define the join operation for finitely many sets so that we can compute n from $\bigoplus_{i<n} X_i$.

Definition 4.10 (Hirschfeldt, Jockusch [24, Definition 4.3])**.** A Turing functional Φ is a *computable strategy* for II for the game reducing P to Q if for all $n \geq 1$, if $Z = \bigoplus_{i<n} X_i$ is the join of Player I's first n moves in some run of said game, then

– if $\Phi^Z(0)\downarrow = 0$, then $\widehat{\Phi}^Z$ is a Z-computable Q-instance;

– otherwise, $\Phi^Z(0)\downarrow = 1$ and $\widehat{\Phi}^Z$ is a Z-computable P-solution to X_0.

We will frequently define Φ^Z by first defining $\widehat{\Phi}^Z$ and then setting $\Phi^Z = \emptyset^\frown\widehat{\Phi}^Z$ or $\Phi^Z = \{0\}^\frown\widehat{\Phi}^Z$.

We say that $P \leq_{gW} Q$ if there is a computable winning strategy for II for the game reducing P to Q. We say that $P \leq_{gW}^n Q$ if there is a computable strategy for II for the game reducing P to Q such that II always wins in round $n+1$ or before.

In this thesis, we will not discuss $\leq_{gW}$, only its bounded versions $\leq_{gW}^n$. In order to understand $\leq_{gW}^n$ better, we start by considering $\leq_{gW}^1$. If $P \leq_{gW}^1 Q$, that means that there is a strategy Φ for II which wins the game reducing P to Q in round 1 or 2. Those P-instances for which Φ wins in round 1 have uniformly computable solutions, while all other P-instances can be solved by solving some corresponding Q-instance (given by Φ). More precisely, Φ provides a Weihrauch reduction from the restriction of P to those latter instances, to Q. This indicates that $\leq_{gW}^1$ and $\leq_W$ are related. We explore their relationship in the following propositions.

First, the above discussion can be formally stated as follows:

Proposition 4.11. *The following are equivalent:*

– $P \leq_{gW}^1 Q$*;*

– *the domain D of P can be computably partitioned into D_0 and D_1, such that $P \restriction D_0$ is uniformly computable and $P \restriction D_1 \leq_W Q$;*

– *there is some uniformly computable R such that $P \leq_W Q \sqcup R$.*

– $P \leq_W Q \sqcup \mathrm{id}$.

Second, if every P-instance uniformly computes a Q-instance, then we can upgrade a $\leq^1_{gW}$-reduction from P to Q to a $\leq_W$-reduction:

Proposition 4.12. *$P \leq_W Q$ if and only if every P-instance uniformly computes a Q-instance (that is,* $\mathrm{dom}(Q)$ *is Medvedev reducible to* $\mathrm{dom}(P)$*) and $P \leq^1_{gW} Q$.*

Proof. ($\Rightarrow$). Fix Γ and Δ witnessing that $P \leq_W Q$. First, Γ witnesses that every P-instance uniformly computes a Q-instance. Next, we give a strategy Φ witnessing that $P \leq^1_{gW} Q$:

$$\Phi^{X_0} = \emptyset^\frown \Gamma^{X_0}$$
$$\Phi^{X_0 \oplus X_1} = \{0\}^\frown \Delta^{X_0 \oplus X_1}.$$

($\Leftarrow$). Fix a strategy Φ witnessing that $P \leq^1_{gW} Q$, and fix a functional Ξ which takes in any P-instance and computes a Q-instance from it. We define functionals Γ and Δ witnessing that $P \leq_W Q$:

$$\Gamma^{X_0} = \begin{cases} \widehat{\Phi}^{X_0} & \text{if } \Phi^{X_0}(0)\downarrow = 0 \\ \Xi^{X_0} & \text{otherwise} \end{cases}$$

and

$$\Delta^{X_0 \oplus X_1} = \begin{cases} \widehat{\Phi}^{X_0 \oplus X_1} & \text{if } \Phi^{X_0}(0)\downarrow = 0 \\ \widehat{\Phi}^{X_0} & \text{otherwise} \end{cases}.$$

□

Most problems that arise directly from mathematical theorems have computable instances. Such problems are called *pointed* (Brattka, de Brecht, Pauly [5]).

Corollary 4.13. *If Q is pointed, then $P \leq_W Q$ if and only if $P \leq^1_{gW} Q$.*

It is clear that Q is pointed if and only if id $\leq_W Q$. Hence if Q is not pointed, then there is a trivial counterexample to the above Corollary: id $\not\leq_W Q$ yet id $\leq^1_{gW}$ Q. These results clarify a statement in §4.4 of [24], where they claim that $P \leq^1_{gW} Q$ if and only if $P \leq_W Q$.

Moving on to $n \geq 1$, observe that if $P \leq^n_{gW} Q$, then there is a computable strategy for II for the game reducing P to Q which wins in round 1 or round $n+1$. This is because everytime II declares victory in round k for $1 < k < n+1$, II could instead repeatedly play the Q-instance which it played in round 1, and wait until round $n+1$ to declare victory. Using this observation, we obtain

Proposition 4.14. *$P \leq^n_{gW} Q$ if and only if the domain D of P can be computably partitioned into D_0 and D_1, such that*

- *$P \restriction D_0$ is uniformly computable;*
- *there is a strategy for II witnessing that $P \restriction D_1 \leq^n_{gW} Q$ which always wins in round $n+1$.*

Proof. ($\Rightarrow$). Fix a strategy Φ witnessing that $P \leq^n_{gW} Q$. For $i = 0, 1$, define $D_i = \{X \in D : \Phi^X(0)\downarrow = 1 - i\}$. D_0 and D_1 form a computable partition of D. $P \restriction D_0$ is uniformly computable, as witnessed by $\widehat{\Phi}$.

Then, as discussed above, we may modify Φ to give a strategy Ψ which always wins the game reducing $P \restriction D_1$ to Q in round $n+1$.

($\Leftarrow$). Fix a computable partition of D into D_0 and D_1, a functional Ξ which solves $P \restriction D_0$, and a strategy Φ which always wins the game reducing $P \restriction D_1$ to Q in round $n+1$.

We give a strategy for II which witnesses that $P \leq_{gW}^{n} Q$. I starts by playing a P-instance, say X_0. II starts by computing whether X_0 lies in D_0 or D_1. If X_0 lies in D_0, then II applies Ξ to solve X_0 and declares victory. If X_0 lies in D_1, then II follows the strategy Φ to solve X_0 and declare victory in round $n+1$. Either way, II declares victory by round $n+1$. $\square$

Another useful property about $\leq_{gW}^{n}$ is that it is well-defined on Weihrauch degrees, which we show below. Since we only defined the compositional product up to Weihrauch degree, this allows us to make sense of statements such as $P \leq_{gW}^{n} Q * Q$ (such as in Theorem 4.29).

The desired statement follows from the following proposition.

Proposition 4.15. *If $P \leq_{gW}^{m} Q$ with a strategy that always wins in round $m+1$ and $Q \leq_{gW}^{n} R$ with a strategy that always wins in round $n+1$, then $P \leq_{gW}^{mn} R$ with a strategy that always wins in round $mn+1$. If $P \leq_{gW}^{m} Q$ and $Q \leq_{gW}^{n} R$, then $P \leq_{gW}^{mn} R$.*

Proof. To prove the first statement, fix a strategy Φ for $P \leq_{gW}^{m} Q$ which always wins in round $m+1$, and a strategy Ψ for $Q \leq_{gW}^{n} R$ which always wins in round $n+1$. We describe a strategy for $P \leq_{gW}^{mn} R$ which always wins in round $mn+1$. The idea is to play the game G reducing P to R by playing the game G' reducing P to Q, interleaved with m many consecutive games $G_0, \ldots, G_{m-1}$, each reducing Q to R.

Say that in G, I starts by playing a P-instance X_0. Then $\widehat{\Phi}(X_0)$ is a Q-instance, so we simulate a parallel game G' reducing P to Q where I starts by playing X_0 and II responds with $\widehat{\Phi}(X_0)$. In order to come up with a valid response for I in G', we simulate yet another parallel game G_0 reducing Q to R where I starts by playing $\widehat{\Phi}(X_0)$. Then $\widehat{\Psi}(\widehat{\Phi}(X_0))$ is an R-instance, so II plays $\widehat{\Psi}(\widehat{\Phi}(X_0))$ in G (and in G_0).

Next, in G, I responds with some R-solution X_1 to $\widehat{\Psi}(\widehat{\Phi}(X_0))$. We copy that response to G_0. Then $\widehat{\Psi}(\widehat{\Phi}(X_0) \oplus X_1)$ is an R-instance, so II plays it in G (and in G_0).

We continue playing G as above (and simulating G_0) until II wins G_0 and provides a Q-solution Z_0 to $\widehat{\Phi}(X_0)$. At that point we return to simulating G': I can now respond with Z_0.

In G', II responds with the Q-instance $\widehat{\Phi}(X_0 \oplus Z_0)$. In order to simulate I's response in G', we simulate another parallel game G_1 reducing Q to R where I starts by playing $\widehat{\Phi}(X_0 \oplus Z_0)$. Proceed as we did for G_0.

Since Φ always wins in round $m+1$ and Ψ always wins in round $n+1$, the above strategy always wins in round $mn+1$.

The proof of the second statement is similar. □

Corollary 4.16. *$\leq^n_{gW}$ is well-defined up to Weihrauch degree, i.e., if $P_1 \leq_W P_0$, $P_0 \leq^n_{gW} Q_0$, and $Q_0 \leq_W Q_1$, then $P_1 \leq^n_{gW} Q_1$.*

Proof. Use Propositions 4.15 and 4.12. □

4.1.4 Step product

The step product generalizes the composition of multivalued functions. Intuitively, $Q \bullet_\Theta P$ corresponds to invoking P, transforming the result by Θ (allowing Θ access to the original P-instance), and then invoking Q.

Definition 4.17 (Dorais, Dzhafarov, Hirst, Mileti, Shafer [15, section 5.2])**.** Given multivalued functions P and Q and a Turing functional Θ, the multivalued function $Q \bullet_\Theta P$ is defined as follows. A is an instance of $Q \bullet_\Theta P$ if

– A is a P-instance;

– for every P-solution B to A, we have that $\Theta^{A \oplus B}$ is a Q-instance.

In that case, a $(Q \bullet_\Theta P)$-solution to A is a pair (B, C) such that

– B is a P-solution to A;

– C is a Q-solution to $\Theta^{A \oplus B}$.

Note that $Q \bullet_\Theta P$ may very well be the empty multivalued function, but that will not affect any of our results. Note also that if we define Θ to be the projection $A \oplus B \mapsto B$, then $Q \bullet_\Theta P$ is exactly $Q \circ P$.

Many compositions that we encounter in proofs can be thought of as some step product. However, the step product does not satisfy several of the properties one would desire of a product, such as monotonicity. First we give a positive result: in some sense, the step product is monotone in the first coordinate with respect to Weihrauch reducibility.

Proposition 4.18. *Suppose $Q_0 \leq_W Q_1$, Θ is a functional, and P is a multivalued function. Then there is a functional Λ such that $Q_0 \bullet_\Theta P \leq_W Q_1 \bullet_\Lambda P$.*

Proof. We define a functional Λ, and forward and backward functionals witnessing that $Q_0 \bullet_\Theta P \leq_W Q_1 \bullet_\Lambda P$. We will take the forward functional to be the identity.

Fix Γ and Δ witnessing that $Q_0 \leq_W Q_1$. We define Λ such that every $(Q_0 \bullet_\Theta P)$-instance X is also a $(Q_1 \bullet_\Lambda P)$-instance: for every P-solution Y to X, $\Theta(X \oplus Y)$ is a Q_0-instance, so $\Gamma(\Theta(X \oplus Y))$ is a Q_1-instance. Hence we define $\Lambda = \Gamma \circ \Theta$.

Next, for every $(Q_1 \bullet_\Lambda P)$-solution (Y, Z) to X, we have that Y is a P-solution to X and Z is a Q_1-solution to $\Lambda(X \oplus Y) = \Gamma(\Theta(X \oplus Y))$. Hence $\Delta(\Theta(X \oplus Y) \oplus Z)$ is a Q_0-solution to $\Theta(X \oplus Y)$, so $(Y, \Delta(\Theta(X \oplus Y) \oplus Z))$ is a $(Q_0 \bullet_\Theta P)$-solution to X. Therefore, we define the backward functional by

$$X \oplus (Y, Z) \mapsto (Y, \Delta(\Theta(X \oplus Y) \oplus Z)).$$

This completes the proof that $Q_0 \bullet_\Theta P \leq_W Q_1 \bullet_\Lambda P$. □

However, the step product is not monotone (in the above sense) in the second coordinate. (Take $Q = \mathsf{RT}^1_2$, $P_0 = \text{id}$, $P_1 = \text{id} \restriction \{\mathbb{N}\}$. Then $P_0 \leq_W P_1$ but for all Λ, $Q \circ P_0 \not\leq_W Q \bullet_\Lambda P_1$. See Example 4.25 for a more sophisticated example.) We have the following partial positive result:

Proposition 4.19. *Suppose $P_0 \leq_W P_1$, P_1 is a cylinder, Θ is a functional, and Q is a multivalued function. Then there is a functional Λ such that $Q \bullet_\Theta P_0 \leq_{sW} Q \bullet_\Lambda P_1$.*

Proof. Fix Γ and Δ witnessing that $P_0 \leq_W P_1$. Fix Φ and Ψ witnessing that $\text{id} \times P_1 \leq_{sW} P_1$. We define a functional Λ, and forward and backward functionals witnessing that $Q \bullet_\Theta P_0 \leq_W Q \bullet_\Lambda P_1$. We will take the forward functional to be $X \mapsto \Phi(X, \Gamma(X))$.

We define Λ such that for every $(Q \bullet_\Theta P_0)$-instance X, $\Phi(X, \Gamma(X))$ is a $(Q \bullet_\Lambda P_1)$-instance: first note that $\Phi(X, \Gamma(X))$ is a P_1-instance. Next, for every P_1-solution Z to $\Phi(X, \Gamma(X))$, $\Psi(Z)$ is an $(\text{id} \times P_1)$-solution to $(X, \Gamma(X))$; that is, $(\Psi(Z))_0 = X$

and $(\Psi(Z))_1$ is a P_1-solution to $\Gamma(X)$. It follows that $\Delta(\Psi(Z))$ is a P_0-solution to X. Therefore, $\Theta(X \oplus \Delta(\Psi(Z)))$ is a Q-instance. So we define

$$\Lambda(A \oplus Z) = \Theta((\Psi(Z))_0 \oplus \Delta(\Psi(Z))).$$

Now, for every $(Q \bullet_\Lambda P_1)$-solution (Z, W) to $\Phi(X, \Gamma(X))$, we have that Z is a P_1-solution to $\Phi(X, \Gamma(X))$ and W is a Q-solution to $\Lambda(\Phi(X, \Gamma(X)) \oplus Z) = \Theta(X \oplus \Delta(\Psi(Z)))$. Then $(\Delta(\Psi(Z)), W)$ is a $(Q \bullet_\Theta P_0)$-solution to X. Therefore, we define the backward functional by

$$(Z, W) \mapsto (\Delta(\Psi(Z)), W).$$

This completes the proof that $Q \bullet_\Theta P_0 \leq_{sW} Q \bullet_\Lambda P_1$. □

Proposition 4.19 suggests that the class of $Q \bullet_\Theta P$ where P is a cylinder may be well-behaved (see also Lemma 4.6). Note that any multivalued function P is Weihrauch equivalent to a cylinder, for example $\mathrm{id} \times P$.

4.2 Composing a multivalued function with itself

In this section, we study the relationships between the various products for the simplest nontrivial case: two invocations of P. We will see in Theorem 4.22 that the compositional product and the reduction game are equivalent in the case where P is pointed, and the compositional product and the step product can be made equivalent if we modify the second factor in the step product.

We begin by showing that $*$ is always at least as strong as $\bullet_\Theta$.

Proposition 4.20. *For any functional Θ, we have that $Q \bullet_\Theta P \leq_W Q * P$.*

Proof. Define the multivalued function P_0 as follows. Instances of P_0 are instances of $Q \bullet_\Theta P$. (Y, Z) is a solution to the P_0-instance Y if Z is a P-solution to Y.

We have $P_0 \leq_W P$: take the forward functional to be the identity, and define the backward functional by mapping $Y \oplus Z$ to (Y, Z).

Next, define Q_0: its instances are pairs (Y, Z) such that Y is a $Q \bullet_\Theta P$-instance and Z is a P-solution to Y. (Z, W) is a solution to the Q_0-instance (Y, Z) if W is a solution to the Q-instance $\Theta^{Y \oplus Z}$.

We have $Q_0 \leq_W Q$: define the forward functional by mapping (Y, Z) to $\Theta^{Y \oplus Z}$, and define the backward functional by mapping $(Y, Z) \oplus W$ to (Z, W).

Finally, we see that $Q_0 \circ P_0$ is equal to $Q \bullet_\Theta P$, so we are done. $\square$

Next, in order to state our first main result, we need the following definition.

Definition 4.21. Given a multivalued function R, define the multivalued function $\overline{R}$ as follows. Instances of $\overline{R}$ are pairs (X, Y), where X is any set and Y is an R-instance. Z is an $\overline{R}$-solution to (X, Y) if Z is an R-solution to Y.

Note that $\overline{R} \equiv_W R$. Note also that $\overline{R}$ is not a cylinder. Now we prove our first main theorem relating $*$, reduction games, and $\bullet_\Theta$.

Theorem 4.22. *The following are equivalent:*

1. $P \leq_W Q * Q$*;*
2. *there is a strategy for II witnessing that* $P \leq^2_{gW} Q$*, which always wins in the third round, or* P *has empty domain;*
3. *every* P*-instance uniformly computes a* Q*-instance, and*
   $P \leq^2_{gW} Q$*;*

4. *there is a functional* Θ *such that* $P \leq_W Q \bullet_\Theta \overline{Q}$.

Proof. (1) $\Rightarrow$ (2). By Theorem 4.5, since $P \leq_W Q * Q$, there are multivalued functions $Q_0, Q_1 \leq_W Q$ such that $P \leq_W Q_1 \circ Q_0$. We define a strategy Φ for II witnessing that $Q_1 \circ Q_0 \leq^2_{gW} Q$, which always wins in the third round. The desired result then follows from Corollary 4.16. Fix Γ_0 and Δ_0 witnessing that $Q_0 \leq_W Q$. Fix Γ_1 and Δ_1 witnessing that $Q_1 \leq_W Q$.

I begins the game by playing a $(Q_1 \circ Q_0)$-instance, say X. (If the domain of $Q_1 \circ Q_0$ is empty, then the domain of P is empty and we are done.) In particular, note that X is a Q_0-instance. II responds by playing the Q-instance $\Gamma_0(X)$.

I then plays a Q-solution to $\Gamma_0(X)$, say Z. Then $\Delta_0(X \oplus Z)$ is a Q_0-solution to X. Since X is a $(Q_1 \circ Q_0)$-instance, $\Delta_0(X \oplus Z)$ must be a Q_1-instance. Therefore, II responds with the Q-instance $\Gamma_1(\Delta_0(X \oplus Z))$.

Finally, I plays a Q-solution W to $\Gamma_1(\Delta_0(X \oplus Z))$. Then $\Delta_1(\Delta_0(X \oplus Z) \oplus W)$ is a Q_1-solution to $\Delta_0(X \oplus Z)$, which implies that it is a $(Q_1 \circ Q_0)$-solution to X. II declares victory and responds with $\Delta_1(\Delta_0(X \oplus Z) \oplus W)$.

(2) $\Rightarrow$ (3). If P has empty domain, (3) vacuously holds. Otherwise, fix a strategy Φ for II witnessing that $P \leq^2_{gW} Q$ which always wins in the third round. For every P-instance X, $\widehat{\Phi}^X$ is always a Q-instance (because Φ does not win in the first round).

(3) $\Rightarrow$ (4). Fix some Φ witnessing that $P \leq^2_{gW} Q$, and fix some Ξ which computes Q-instances from P-instances. First define a forward functional for

$P \leq_W Q \bullet_\Theta \overline{Q}$:

$$\Gamma^X = \begin{cases} (X, \widehat{\Phi}^X) & \text{if } \Phi^X(0)\!\downarrow = 0 \\ (X, \Xi^X) & \text{otherwise} \end{cases}.$$

Then define

$$\Theta^{(X,Y)\oplus Z} = \begin{cases} \widehat{\Phi}^{X\oplus Z} & \text{if } \Phi^X(0)\!\downarrow = 0 \text{ and } \Phi^{X\oplus Z}(0)\!\downarrow = 0 \\ \Xi^X & \text{otherwise} \end{cases}.$$

Observe that for every P-instance X, Γ^X is a $\overline{Q}$-instance, and for every $\overline{Q}$-solution Z to Γ^X, $\Theta^{\Gamma^X \oplus Z}$ is a Q-instance. Therefore Γ^X is a $Q \bullet_\Theta \overline{Q}$-instance.

Finally, define a backward functional

$$\Delta^{X\oplus(Z,W)} = \begin{cases} \widehat{\Phi}^{X\oplus Z\oplus W} & \text{if } \Phi^X(0)\!\downarrow = 0 \text{ and } \Phi^{X\oplus Z}(0)\!\downarrow = 0 \\ \widehat{\Phi}^{X\oplus Z} & \text{if } \Phi^X(0)\!\downarrow = 0 \text{ and } \Phi^{X\oplus Z}(0)\!\downarrow = 1 \\ \widehat{\Phi}^X & \text{if } \Phi^X(0)\!\downarrow = 1 \end{cases}.$$

<u>(4) ⇒ (1).</u> We have that

$$\begin{aligned} P &\leq_W Q \bullet_\Theta \overline{Q} & \\ &\leq_W Q * \overline{Q} & \text{Proposition 4.20} \\ &\leq_W Q * Q & \overline{Q} \leq_W Q \text{ and definition of } * . \end{aligned}$$

□

We note that a statement similar to (2) ⇒ (4) was proven in Remark 4.23 in [24]. (They use $\widehat{Q} := \mathrm{id} \times Q$ instead of $\overline{Q}$, but the same result holds: use Proposition 4.19 and the fact that $\widehat{Q}$ is a cylinder.) However, they (implicitly) assume that if $P \leq^2_{gW} Q$, then (2) holds. This is true if Q is pointed, but false otherwise (see Proposition 4.27).

Let us now study corollaries of Theorem 4.22. First, we obtain a simple realization of the compositional product (cf. Theorem 4.5):

Corollary 4.23. *For all Q, there is a functional Θ such that $Q * Q \equiv_W Q \bullet_\Theta \overline{Q}$.*

Proof. (1) $\Leftrightarrow$ (4) in Theorem 4.22. □

If Q is a cylinder, we note that a nicer result follows from the cylindrical decomposition of Brattka and Pauly (Lemma 4.6):

Corollary 4.24. *If Q is a cylinder, then there is a functional Θ such that $Q*Q \equiv_W Q \bullet_\Theta Q$.*

Proof. By the cylindrical decomposition lemma, there is some uniformly computable K such that $Q * Q \equiv_W Q \circ K \circ Q$. Taking $\Theta : A \oplus B \mapsto K(B)$, we get $Q * Q \equiv_W Q \bullet_\Theta Q$. □

The above corollary cannot hold for all Q in general:

Example 4.25. We construct Q and Θ such that for all Λ, $Q \bullet_\Theta \overline{Q} \not\leq_W Q \bullet_\Lambda Q$ (and hence $Q * Q \not\leq_W Q \bullet_\Lambda Q$ for all Λ). We take Θ to be the identity. Fix four sets A, B, C and D such that no three of these sets compute the other. (Such sets can be obtained from a Cohen generic.) Define Q as follows: the instance B has a unique solution C, and the instance $((A, B), C)$ has a unique solution D. Observe that (A, B) is a $(Q \bullet_{\mathrm{id}} \overline{Q})$-instance with unique solution (C, D).

Suppose towards a contradiction that Λ is such that $Q \bullet_{\mathrm{id}} \overline{Q} \leq_W Q \bullet_\Lambda Q$. Fix Γ and Δ witnessing this. We show that they fail to solve the $(Q \bullet_{\mathrm{id}} \overline{Q})$-instance (A, B). First, $\Gamma(A \oplus B)$ must be a Q-instance. The only Q-instance computable in $A \oplus B$ is B, which has a unique Q-solution C. Next, $\Lambda(B \oplus C)$ must be a

Q-instance. The only Q-instance computable in $B \oplus C$ is B, which has a unique Q-solution C. Hence the unique $(Q \bullet_\Lambda Q)$-solution to B must be (C, C). Finally, $\Delta((A \oplus B) \oplus (C \oplus C))$ must be the unique $(Q \bullet_{\mathrm{id}} \overline{Q})$-solution to (A, B), which is (C, D). But $A \oplus B \oplus C$ does not compute D, contradiction.

Another application of Theorem 4.22 is to compare $\bullet$, $*$, and $\leq^2_{gW}$ on the same footing. The following suprema are taken with respect to Weihrauch reducibility.

Theorem 4.26. *For all Q, $\sup_\Lambda Q \bullet_\Lambda \overline{Q}$ exists and for all Θ,*

$$Q \bullet_\Theta Q \leq_W \sup_\Lambda Q \bullet_\Lambda \overline{Q} \equiv_W Q * Q \leq^2_{gW} Q.$$

Proof. First, by (1) $\Rightarrow$ (4) in Theorem 4.22, there is Λ such that $Q * Q \leq_W Q \bullet_\Lambda \overline{Q}$. By (4) $\Rightarrow$ (1) in Theorem 4.22, $Q \bullet_\Lambda \overline{Q} \leq Q * Q$ for all Λ. Hence $\sup_\Lambda Q \bullet_\Lambda \overline{Q}$ exists and is equal to $Q * Q$.

Next, by Proposition 4.20, $Q \bullet_\Theta Q \leq_W Q * Q$.

Finally, by (1) $\Rightarrow$ (2) in Theorem 4.22, $Q * Q \leq^2_{gW} Q$. □

We do not know whether $\sup_\Theta Q \bullet_\Theta Q$ or $\sup\{P : P \leq^2_{gW} Q\}$ exist in general. If Q is pointed, we have some partial results.

Proposition 4.27. *If Q is pointed, then $\sup\{P : P \leq^2_{gW} Q\}$ exists and is equal to $Q * Q$. If Q is not pointed, then there is some $P \leq^2_{gW} Q$ (in fact, $P \leq^1_{gW} Q$) such that $P \not\leq_W Q * Q$.*

Proof. Suppose that Q has a computable instance. If we fix a computable Q-instance A, then for every multivalued function P, every P-instance uniformly computes A. By (1) $\Leftrightarrow$ (3) in Theorem 4.22, $\sup\{P : P \leq_{gW}^2 Q\}$ exists and is equal to $Q * Q$.

Suppose that Q has no computable instance. Consider $P = \text{id}$. We have that $P \leq_{gW}^1 Q$, yet P-instances do not uniformly compute Q-instances. By the contrapositive of (1) $\Rightarrow$ (3) in Theorem 4.22, $P \not\leq_W Q * Q$. $\square$

Corollary 4.28. *If Q is pointed, then*

$$\sup_{\Lambda} Q \bullet_{\Lambda} \overline{Q} \equiv_W Q * Q \equiv_W \sup\{P : P \leq_{gW}^2 Q\}.$$

Proposition 4.27 inspired us to consider $\leq_{gW}^1$ instead of $\leq_W$. That gives us a cleaner analog of Theorem 4.22:

Theorem 4.29. *The following are equivalent:*

1. $P \leq_{gW}^1 Q * Q$;
2. $P \leq_{gW}^2 Q$;
3. *there is a functional Θ such that $P \leq_{gW}^1 Q \bullet_{\Theta} \overline{Q}$.*

Proof. (1) $\Rightarrow$ (2). Let D be the domain of P. By Proposition 4.11, fix a computable partition D_0 and D_1 of D such that $P \restriction D_0$ is uniformly computable and $P \restriction D_1 \leq_W Q * Q$. By (1) $\Rightarrow$ (2) in Theorem 4.22, there is a strategy for II witnessing that $P \restriction D_1 \leq_{gW}^2 Q$, which always wins in the third round. By Proposition 4.14, $P \leq_{gW}^2 Q$ as desired.

(2) $\Rightarrow$ (3). Let D be the domain of P. By Proposition 4.14, fix a computable partition D_0 and D_1 of D such that $P \restriction D_0$ is uniformly computable, and there exists a strategy Φ witnessing that $P \restriction D_1 \leq_{gW}^2 Q$ which always wins in the third

round. By (2) ⇒ (4) in Theorem 4.22, there is some Θ such that $P \restriction D_1 \leq_W Q \bullet_\Theta \overline{Q}$. By Proposition 4.11, $P \leq^1_{gW} Q \bullet_\Theta \overline{Q}$ as desired.

(3) ⇒ (1). By Theorem 4.26, $Q \bullet_\Theta \overline{Q} \leq_W Q * Q$. The desired result follows from Corollary 4.16. □

4.3 Finite compositions of arbitrary multivalued functions

Many of the results in Section 4.2 can be easily generalized to finite compositions of a multivalued function with itself. In this section, we generalize some of our results to the finite composition of (possibly) different multivalued functions. We show that such a composition can be thought of in terms of the following generalized reduction game.

Definition 4.30. For multivalued functions P, $Q_0, \ldots, Q_{n-1}$, define the *game reducing P to $Q_{n-1}, \ldots, Q_0$* as follows. In round 1, Player I starts by playing a P-instance X_0. Player II responds with either of the following:

- an X_0-computable P-solution to X_0;
- an X_0-computable Q_0-instance Y_1;

and an indication of which case it is (for the first case, II declares victory.)

Subsequently, for $k \geq 1$, in round $k+1$, Player I plays a solution X_k to the Q_{k-1}-instance Y_k. Player II responds with either of the following:

- a $\left(\bigoplus_{i<k+1} X_i\right)$-computable P-solution to X_0;
- if $k < n$, a $\left(\bigoplus_{i<k+1} X_i\right)$-computable Q_k-instance Y_{k+1};

and an indication of which case it is (for the first case, II declares victory.)

Player II wins if it declares victory on round $n+1$ or before, after which the game ends. Otherwise Player I wins, which happens exactly if Player II has no possible move in some round. (If the game reaches round $n+1$, the only possible move for II is to declare victory, if it can.)

Note. In the game reducing P to Q, if II was able to make a move in round 1, then it can repeat said move for all subsequent rounds. This is not always possible for the game reducing P to $Q_{n-1}, \ldots, Q_0$.

Definition 4.31. A Turing functional Φ is a *computable strategy* for II for the game reducing P to $Q_{n-1}, \ldots, Q_0$ if for all $k \leq n$, if $Z = \bigoplus_{i<k+1} X_i$ is the join of Player I's first $k+1$ moves in some run of said game, then $\Phi^Z = V^\frown Y$, where

- if $V = \{0\}$, then Y is a Z-computable solution to the P-instance X_0 (this must happen if $k = n$);
- otherwise, $V = \emptyset$ and Y is a Z-computable Q_k-instance.

We define $\widehat{\Phi}$ and the join operation as before.

We say that $P \leq_{gW}^{(n)} Q_{n-1}, \ldots, Q_0$ if there is a computable winning strategy for II for the game reducing P to $Q_{n-1}, \ldots, Q_0$.

Unlike $\leq_{gW}^{n}$, $\leq_{gW}^{(n)}$ does not seem to admit a nice characterization like that in Proposition 4.14. That is, assuming that $P \leq_{gW}^{(n)} Q_{n-1}, \ldots, Q_0$, one may not be able to divide the domain of P into finitely many sets, on each of which II has a strategy which always wins in a certain number of rounds. Take for example a run where a strategy Φ wins the game reducing P to $Q_{n-1}, \ldots, Q_0$ in some round

$1 < k < n+1$. We may not be able to delay Φ's victory because there may not be any Q_{k+1}-instance which is computable in I's plays. Even if there is such a Q_{k+1}-instance, we may not be able to compute it uniformly from I's plays. Whether we can do so may depend on I's choice of solutions to the instances played by II. Therefore, we do not have an analog of Theorem 4.29 in this context.

Next, we prove an analog of Corollary 4.16. We could prove an analog of Proposition 4.15 and use that to derive an analog of Corollary 4.16, but that would be messy.

Proposition 4.32. *Suppose $P_0 \leq_W P_1$ and $Q_i \leq_W R_i$ for each $i < n$. If $P_1 \leq_{gW}^{(n)} Q_{n-1}, \ldots, Q_0$, then $P_0 \leq_{gW}^{(n)} R_{n-1}, \ldots, R_0$. Moreover, if $P_1 \leq_{gW}^{(n)} Q_{n-1}, \ldots, Q_0$ with a strategy that always wins in the last round, then $P_0 \leq_{gW}^{(n)} R_{n-1}, \ldots, R_0$ with a strategy that always wins in the last round as well.*

Proof. Fix Γ and Δ witnessing that $P_0 \leq_W P_1$, and for each $i < n$, fix Γ_i and Δ_i witnessing that $Q_i \leq_W R_i$. Fix a strategy Φ witnessing that $P_1 \leq_{gW}^{(n)} Q_{n-1}, \ldots, Q_0$. We describe a strategy Ψ witnessing that $P_0 \leq_{gW}^{(n)} R_{n-1}, \ldots, R_0$, such that if Φ always wins in round $n+1$, then so does Ψ. The idea is as follows: while we play the game G_0 reducing P_0 to $R_{n-1}, \ldots, R_0$, we play a parallel game G_1 reducing P_1 to $Q_{n-1}, \ldots, Q_0$, where II follows the strategy Φ.

In the game G_0, I starts by playing a P_0-instance X_0. Then $\Gamma(X_0)$ is a P_1-instance, so we may start the game G_1 with the P_1-instance $\Gamma(X_0)$ and with II following the strategy Φ. In G_1, II either plays a P_1-solution to $\Gamma(X_0)$ and declares victory, or a Q_0-instance.

If II plays a P_1-solution to $\Gamma(X_0)$, then we may apply Δ to obtain a P_0-solution to X_0. II can then play this set in G_0 and declare victory.

On the other hand, if II plays a Q_0-instance, then we may apply Γ_0 to obtain an R_0-instance. II can then play this set in G_0, continuing the game.

In G_0 (if II has not already won), I responds by playing an R_0-solution to II's previous play in G_0. Then we may apply Δ_0 to obtain a Q_0-solution to II's previous play in G_1. We make I play this set in G_1.

Next, in G_1, II (following Φ) either plays a P_1-solution to $\Gamma(X_0)$ and declares victory, or plays a Q_1-instance. The rest of the game proceeds as above.

We have described our strategy for the first two rounds of G_0. We omit the formal construction and verification. □

Our final main theorem (analogous to Theorem 4.22) is as follows:

Theorem 4.33. *For multivalued functions $P, Q_{n-1}, \dots, Q_0$, the following are equivalent:*

1. $P \leq_W Q_{n-1} * \cdots * Q_0$;
2. *there is a strategy for II witnessing that $P \leq_{gW}^{(n)} Q_{n-1}, \dots, Q_0$ which always wins in round $n+1$, or P has empty domain;*
3. *there are functionals $\Theta_0, \dots, \Theta_{n-2}$ such that*
   $P \leq_W \overline{Q_{n-1}} \bullet_{\Theta_{n-2}} (\overline{Q_{n-2}} \bullet_{\Theta_{n-3}} (\cdots (\overline{Q_1} \bullet_{\Theta_0} \overline{Q_0})))$.

Before we give the proof, we state some observations. First, if all Q_i are pointed, then the extra condition in (2) is unnecessary (cf. the observation before Proposition 4.14):

Corollary 4.34. *For multivalued functions $P, Q_{n-1}, \dots, Q_0$ such that P has nonempty* *domain and all Q_i are pointed, $P \leq_W Q_{n-1} * \cdots * Q_0$ if and only if $P \leq_{gW}^{(n)} Q_{n-1}, \ldots, Q_0$.*

Proof. (⇒) follows from (1) ⇒ (2) in Theorem 4.33. For (⇐), fix computable instances of each Q_i. Then any strategy witnessing that $P \leq_{gW}^{(n)} Q_{n-1}, \ldots, Q_0$ can be padded to obtain a strategy which always wins in the last round: simply play the appropriate computable instances and ignore the solutions. Then apply (2) ⇒ (1) in Theorem 4.33. □

Unlike Proposition 4.27, even if for all P, we have $P \leq_W Q_{n-1} * \cdots * Q_0$ if and only if $P \leq_{gW}^{(n)} Q_{n-1}, \ldots, Q_0$, it does not follow that all Q_i have computable instances. (See the comments before Proposition 4.32.)

Next, note that strategies in the game reducing P to $Q_{n-1}, \ldots, Q_0$ are allowed to refer to each Q_i-instance played thus far, while $\bullet_\Theta$ only allows reference to the Q_i-instance just played. Therefore in (3), we use $\overline{Q_i}$ instead of Q_i. The extra coordinate in a $\overline{Q_i}$-instance can be used to encode every Q_j-instance played thus far. For the last factor ($i = n-1$), we can get away with Q_{n-1} instead of $\overline{Q_{n-1}}$ (as is the case in Theorem 4.22). Nevertheless, we state the theorem with $\overline{Q_{n-1}}$ because this obviates the need to consider an extra case in the proof of (2) ⇒ (3).

We now prove Theorem 4.33:

Proof. <u>(1) ⇒ (2).</u> By Lemma 4.8, since $P \leq_W Q_{n-1} * \cdots * Q_0$, there are multivalued functions $R_0, \ldots, R_{n-1}$ such that $R_i \leq_W Q_i$ for all $i < n$, and $P \leq_W R_{n-1} \circ \cdots \circ R_0$.

By Proposition 4.32, it suffices to give a computable strategy for II which always wins the game reducing $R_{n-1} \circ \cdots \circ R_0$ to $Q_{n-1}, \ldots, Q_0$ in round $n+1$. For each $i < n$, fix Γ_i and Δ_i witnessing that $R_i \leq_W Q_i$.

In order to illustrate the construction, we describe the strategy for the first three rounds before giving the general description. I starts by playing an $(R_{n-1} \circ \cdots \circ R_0)$-instance X_0. (If $R_{n-1} \circ \cdots \circ R_0$ has empty domain, then so does P and we are done.) II has to respond with an X_0-computable Q_0-instance. Note that X_0 is in particular an R_0-instance, so II can play the Q_0-instance $\Gamma_0(X_0)$.

Next, I plays a Q_0-solution X_1 to $\Gamma_0(X_0)$. II has to respond with an $(X_0 \oplus X_1)$-computable Q_1-instance. Since X_0 is an $(R_{n-1} \circ \cdots \circ R_0)$-instance, any R_0-solution to X_0 is itself an $(R_{n-1} \circ \cdots \circ R_1)$-instance, which is in particular an R_1-instance. We can obtain an R_0-solution to X_0 by applying Δ_0 to $X_0 \oplus X_1$. As explained above, that gives us an R_1-instance, to which we can apply Γ_1 to obtain a Q_1-instance. Therefore II plays $\Gamma_1(\Delta_0(X_0 \oplus X_1))$.

In the third round, I plays a Q_1-solution X_2 to $\Gamma_1(\Delta_0(X_0 \oplus X_1))$. II has to respond with an $(X_0 \oplus X_1 \oplus X_2)$-computable Q_2-instance.

Since $\Delta_0(X_0 \oplus X_1)$ is an $(R_{n-1} \circ \cdots \circ R_1)$-instance, any R_1-solution to $\Delta_0(X_0 \oplus X_1)$ is itself an $(R_{n-1} \circ \cdots \circ R_2)$-instance, which is in particular an R_2-instance. We can obtain an R_1-solution to $\Delta_0(X_0 \oplus X_1)$ by applying Δ_1 to $\Delta_0(X_0 \oplus X_1) \oplus X_2$. That gives us an R_2-instance, to which we can apply Γ_2 to obtain a Q_2-instance. Therefore II plays $\Gamma_2(\Delta_1(\Delta_0(X_0 \oplus X_1) \oplus X_2))$.

We have described our strategy for the first three rounds. Formally, define the auxiliary functional Ξ by recursion:

$$\begin{aligned} \Xi(X_0) &= X_0 \\ \Xi\left(\bigoplus_{j<k+1} X_j\right) &= \Delta_{k-1}\left(\Xi\left(\bigoplus_{j<k} X_j\right) \oplus X_k\right) \qquad \text{if } k \leq n. \end{aligned}$$

For example, $\Xi(X_0 \oplus X_1) = \Delta_0(X_0 \oplus X_1)$. Then we can define our strategy as follows. Suppose that in round k, I plays X_{k-1}. In round $k < n+1$, II plays the Q_{k-1}-instance $\Gamma_{k-1}(\Xi(\bigoplus_{j<k} X_j))$. In round $n+1$, II declares victory and plays $\Xi(\bigoplus_{j<n+1} X_j)$.

Verification. We show by simultaneous induction on k that:

(i) for every $1 \leq k < n+1$, $\Xi(\bigoplus_{j<k} X_j)$ is an $(R_{n-1} \circ \cdots \circ R_{k-1})$-instance;

(ii) for every $1 \leq k \leq n+1$, II's move in round k is legal;

(iii) for every $1 < k \leq n+1$, $\Xi(\bigoplus_{j<k} X_j)$ is an R_{k-2}-solution to the $(R_{n-1} \circ \cdots \circ R_{k-2})$-instance $\Xi(\bigoplus_{j<k-1} X_j)$.

Base case. By definition of Ξ and the game, (i) holds for $k = 1$.

Inductive step 1. Suppose (i) holds for some $1 \leq k < n+1$. Then $\Xi(\bigoplus_{j<k} X_j)$ is in particular an R_{k-1}-instance, so by choice of Γ_{k-1}, II's move in round k is a Q_{k-1}-instance. Also, $\Gamma_{k-1} \circ \Xi$ is computable. Therefore (ii) holds for k.

Inductive step 2. Suppose (i) and (ii) hold for some $1 \leq k < n+1$. Then in round $k+1$, I plays a solution X_k to II's move in round k. By our choice of Δ_{k-1} and the definition of Ξ, (iii) holds for $k+1$.

Inductive step 3. Suppose (iii) holds for some $1 < k < n+1$. By definition of $\circ$, (i) is true for k as well.

The base case and inductive steps prove (i), (ii), and (iii) for the desired values of k, except (ii) for $k = n+1$. We prove that as follows. Since (iii) holds for every $1 < k \leq n+1$, by definition of $\circ$, $\Xi(\bigoplus_{j<n+1} X_j)$ is a $(R_{n-1} \circ \cdots \circ R_0)$-solution to X_0. Therefore $\Xi(\bigoplus_{j<n+1} X_j)$ is a winning move for II in round $n+1$. We have defined a strategy for II which always wins the game reducing P to $Q_{n-1}, \ldots, Q_0$

in round $n+1$.

(2) $\Rightarrow$ (3). If P has empty domain, (3) vacuously holds. Otherwise, fix a strategy Φ for II which always wins the game reducing P to $Q_{n-1}, \dots, Q_0$ in round $n+1$. We have to define $\Theta_0, \dots, \Theta_{n-2}$ and forward and backward functionals witnessing that $P \leq_W \overline{Q_{n-1}} \bullet_{\Theta_{n-2}} (\overline{Q_{n-2}} \bullet_{\Theta_{n-3}} (\cdots (\overline{Q_1} \bullet_{\Theta_0} \overline{Q_0})))$.

Suppose we are given a P-instance X_0, from which we need to compute a $\overline{Q_{n-1}} \bullet_{\Theta_{n-2}} (\overline{Q_{n-2}} \bullet_{\Theta_{n-3}} (\cdots (\overline{Q_1} \bullet_{\Theta_0} \overline{Q_0})))$-instance. Regardless of our definitions of $\Theta_0, \dots, \Theta_{n-2}$, such a set must be a $\overline{Q_0}$-instance. As a starting point, we can obtain a Q_0-instance by applying $\widehat{\Phi}$ to X_0. Also, we want to include X_0 in the $\overline{Q_0}$-instance so that we can use it in the future. Hence, we define the forward functional Γ to send X_0 to the $\overline{Q_0}$-instance $(X_0, \widehat{\Phi}(X_0))$.

Next, we need to define Θ_0 so that for every $\overline{Q_0}$-solution X_1 to $(X_0, \widehat{\Phi}(X_0))$, $\Theta_0((X_0, \widehat{\Phi}(X_0)) \oplus X_1)$ is a $\overline{Q_1}$-instance. Since X_1 is a Q_0-solution to $\widehat{\Phi}(X_0)$, we can obtain a Q_1-instance by applying $\widehat{\Phi}$ to $X_0 \oplus X_1$. Also, we want to include X_0 and X_1 in the $\overline{Q_1}$-instance so that we can use them in the future. Hence, we define Θ_0 to output the $\overline{Q_1}$-instance $(X_0 \oplus X_1, \widehat{\Phi}(X_0 \oplus X_1))$.

In general, for $0 \leq m \leq n-2$, define Θ_m by

$$(X_0, \widehat{\Phi}(X_0)) \oplus (((X_1, X_2), \dots), X_{m+1}) \mapsto \left(\bigoplus_{i<m+2} X_i, \widehat{\Phi}\left(\bigoplus_{i<m+2} X_i \right) \right).$$

Finally, we want to solve X_0 using a $\overline{Q_{n-1}} \bullet_{\Theta_{n-2}} (\cdots (\overline{Q_1} \bullet_{\Theta_0} \overline{Q_0}))$-solution to $(X_0, \widehat{\Phi}(X_0))$. Such a solution has the form $(((X_1, X_2), \dots), X_n)$. We will show in the verification that there is a run of the game reducing P to $Q_{n-1}, \dots, Q_0$ where II follows the strategy Φ and at round m, I plays X_{m-1}. Since Φ always wins in round $n+1$, $\widehat{\Phi}(\bigoplus_{i<n+1} X_i)$ must be a P-solution to X_0. Therefore, we define the

backward functional Δ by mapping $X_0 \oplus (((X_1, X_2), \dots), X_n)$ to $\widehat{\Phi}(\bigoplus_{i<n+1} X_i)$.

Verification. We show that $P \leq_W \overline{Q_{n-1}} \bullet_{\Theta_{n-2}} (\overline{Q_{n-2}} \bullet_{\Theta_{n-3}} (\cdots (\overline{Q_1} \bullet_{\Theta_0} \overline{Q_0})))$ via Γ and Δ. Fix a P-instance X_0. We show by simultaneous induction on k that

(i) for each $0 \leq k \leq n-1$, $\Gamma(X_0)$ is a $\overline{Q_k} \bullet_{\Theta_{k-1}} (\cdots (\overline{Q_1} \bullet_{\Theta_0} \overline{Q_0}))$-instance;

(ii) for each $0 \leq k \leq n-1$, if $(((X_1, X_2), \dots), X_{k+1})$ is a $\overline{Q_k} \bullet_{\Theta_{k-1}} (\cdots (\overline{Q_1} \bullet_{\Theta_0} \overline{Q_0}))$-solution to $\Gamma(X_0)$, then there is a partial run of the game reducing P to $Q_{n-1}, \dots, Q_0$ where II follows the strategy Φ and at round $1 \leq m \leq k+2$, I plays X_{m-1}.

Base case. We show that (i) holds for $k = 0$. Since X_0 is a P-instance and Φ always wins in round $n+1$, it follows that $\widehat{\Phi}(X_0)$ is a Q_0-instance. Therefore $\Gamma(X_0) = (X_0, \widehat{\Phi}(X_0))$ is a $\overline{Q_0}$-instance.

Inductive step 1. Assuming that for some $0 \leq k \leq n-1$, we have that (ii) holds for all $0 \leq m < k$ and (i) holds for k, we show that (ii) holds for k. Let $(((X_1, X_2), \dots), X_{k+1})$ be a $\overline{Q_k} \bullet_{\Theta_{k-1}} (\cdots (\overline{Q_1} \bullet_{\Theta_0} \overline{Q_0}))$-solution to $\Gamma(X_0)$. We start by showing that there is a partial run where II follows the strategy Φ and at round $1 \leq m \leq k+1$, I plays X_{m-1}.

If $k = 0$, then I starts by playing the P-instance X_0. If $k > 0$, by definition of $\bullet$, $(((X_1, X_2), \dots), X_k)$ is a $\overline{Q_{k-1}} \bullet_{\Theta_{k-2}} (\cdots (\overline{Q_1} \bullet_{\Theta_0} \overline{Q_0}))$-solution to $\Gamma(X_0)$. By assumption, (ii) holds for $k-1$, so there is a partial run where II follows the strategy Φ and at round $1 \leq m \leq k+1$, I plays X_{m-1}.

Now, we extend said partial run. By choice of $(((X_1, X_2), \dots), X_{k+1})$ and definition of $\bullet$, X_{k+1} is a $\overline{Q_k}$-solution to $\Theta_{k-1}(\Gamma(X_0) \oplus (((X_1, X_2), \dots), X_k))$, which is defined to be $(\bigoplus_{i<k+1} X_i, \widehat{\Phi}(\bigoplus_{i<k+1} X_i))$. Therefore X_{k+1} is a Q_k-solution to

$\widehat{\Phi}(\bigoplus_{i<k+1} X_i)$, and so we may extend the aforementioned run by making I play X_{k+1}. This proves that (ii) holds for k.

Inductive step 2. Assuming that (i) and (ii) hold for some $0 \leq k < n-1$, we show that (i) holds for $k+1$. Since (i) holds for k, it remains to show that if $(((X_1, X_2), \dots), X_{k+1})$ is a $\overline{Q_k} \bullet_{\Theta_{k-1}} (\cdots (\overline{Q_1} \bullet_{\Theta_0} \overline{Q_0}))$-solution to $\Gamma(X_0)$, then $\Theta_k(\Gamma(X_0) \oplus (((X_1, X_2), \dots), X_{k+1})) = (\bigoplus_{i<k+2} X_i, \widehat{\Phi}(\bigoplus_{i<k+2} X_i))$ is a $\overline{Q_{k+1}}$-instance.

Indeed, let us apply (ii) for k to $(((X_1, X_2), \dots), X_{k+1})$. Since Φ always wins in round $n+1$ and $k+2 < n+1$, we have that $\widehat{\Phi}(\bigoplus_{i<k+2} X_i)$ is a Q_{k+1}-instance. We have shown that (i) holds for $k+1$, completing the proof of inductive step 2.

Applying the above base case and inductive steps, we may deduce (i) and (ii) for $k = n-1$. To complete the proof, we show that if $(((X_1, X_2), \dots), X_n)$ is a $\overline{Q_{n-1}} \bullet_{\Theta_{n-2}} (\cdots (\overline{Q_1} \bullet_{\Theta_0} \overline{Q_0}))$-solution to $\Gamma(X_0)$, then $\Delta(X_0 \oplus (((X_1, X_2), \dots), X_n)) = \widehat{\Phi}(\bigoplus_{i<n+1} X_i)$ is a P-solution to X_0.

By (ii) for $k = n-1$, there is a partial run where II follows the strategy Φ and at round $1 \leq m \leq n+1$, I plays X_{m-1}. Since Φ wins in round $n+1$, $\widehat{\Phi}(\bigoplus_{i<n+1} X_i)$ is a P-solution to X_0 as desired.

(3) $\Rightarrow$ (1). Induction on n using Proposition 4.20. □

4.4 The $\equiv^1_{gW}$-lattice

Recall from Proposition 4.11 that $P \leq^1_{gW} Q$ if and only if $P \leq_W Q \sqcup \mathrm{id}$. It follows that $\leq^1_{gW}$ is reflexive and transitive, so we can define the associated notion of $\equiv^1_{gW}$

and $\equiv^1_{gW}$-degrees. As a notion of reduction between problems, we find $\leq^1_{gW}$ more intuitive than $\leq_W$. This is because in order to show that $P \leq_W Q$, one is obliged to compute a Q-instance from every P-instance, even if one could already compute a solution to said P-instance. See also Theorem 4.29.

Using Proposition 4.11, it is easy to show that the $\equiv^1_{gW}$-degrees form a distributive lattice with the usual join and meet operations. In fact, Pauly[5] has pointed out that the $\equiv^1_{gW}$-lattice is isomorphic to the pointed Weihrauch lattice, which was studied by Higuchi and Pauly [23]. It is easy to show that the pointed Weihrauch degrees (under $\leq_W$) form a lattice under the usual join and meet operations.

Proposition 4.35 (Pauly)**.** *The $\equiv^1_{gW}$-lattice and the pointed Weihrauch lattice are isomorphic.*

Proof. By Proposition 4.11, $P \leq^1_{gW} Q$ if and only if $P \leq_W Q \sqcup \mathrm{id}$. Also, it is easy to see that $P \leq_W Q \sqcup \mathrm{id}$ if and only if $P \sqcup \mathrm{id} \leq_W Q \sqcup \mathrm{id}$. Next, note that if P is pointed, then $P \sqcup \mathrm{id} \equiv_W P$. So $P \mapsto P \sqcup \mathrm{id}$ is an isomorphism between the $\equiv^1_{gW}$-degrees and the pointed Weihrauch degrees. Hence $P \mapsto P \sqcup \mathrm{id}$ is in fact a lattice isomorphism. □

[5] Arno Pauly, personal communication.

CHAPTER 5

PARALLEL PRODUCTS OF THE INFINITE PIGEONHOLE PRINCIPLE

This chapter is part of joint work with Dzhafarov, Hirschfeldt, Patey, and Pauly [17], which will appear in Computability.

In this chapter, we investigate the infinite pigeonhole principle for different numbers of colors, and how these problems behave under Weihrauch reducibility with respect to parallel products. Let RT^1_k denote the following problem: given a coloring $c : \mathbb{N} \to k$, produce an infinite c-homogeneous set. A motivating toy example is the fact that $\mathsf{RT}^1_2 \times \mathsf{RT}^1_2 \leq_W \mathsf{RT}^1_4$. More generally, it is easy to see that for all $n \geq 1$ and $k_0, \ldots, k_n \geq 2$,

$$\prod_{m=0}^{n} \mathsf{RT}^1_{k_m} \leq_{sW} \mathsf{RT}^1_{\prod_{m=0}^{n} k_m}.$$

We show below that the right-hand side is optimal. Our results extend a number of similar investigations, including by Dorais, Dzhafarov, Hirst, Mileti and Shafer [15], Hirschfeldt and Jockusch [24], and Patey [34].

In the sequel, we will regard RT^1_k as the problem whose instances are colorings $c : \mathbb{N} \to k$ and whose solutions are colors which appear infinitely often in c. Note that this formulation of RT^1_k is Weihrauch equivalent to the above formulation.

5.1 The product coloring is optimal

We begin with the following technical lemma:

Lemma 5.1. *Suppose that $P \leq_W Q$ and they satisfy the following properties:*

– *P has finite tolerance, i.e., there is some Θ such that if C_0 and C_1 are P-instances, $C_0(x) = C_1(x)$ for all x above some m, and S_0 is a P-solution to C_0, then $\Theta(S_0 \oplus m)$ is a P-solution to C_1;*

– *any finite modification of a P-instance is still a P-instance;*

– *solutions of all instances of P and Q lie in some fixed finite set.*

Then $P \leq_{sW} Q$.

Proof. Fix functionals Φ and Ψ witnessing that $P \leq_W Q$. Since solutions of all instances of P lie in some fixed finite set, we may assume that for each P-instance C and each s which is a Q-solution to $\Phi(C)$, $\Psi(C \oplus s)$ outputs a number which codes a P-solution to C. Fix a functional Θ witnessing that P has finite tolerance. Fix a finite solution set S for Q. We define functionals which witness that $P \leq_{sW} Q$.

First, we construct a finite initial segment τ of a P-instance C which decides (in the sense of Cohen 1-genericity) for each $s \in S$ whether $\Psi(C \oplus s)$ converges. Since S is finite, such τ exists.

We define Φ' by $\Phi'(C) = \Phi(C')$, where C' is obtained from C by replacing its initial segment of length $|\tau|$ by τ itself. By our assumption on P, C' is still a P-instance.

We define Ψ' by $\Psi'(s) = \Theta(\Psi(\tau \oplus s) \oplus |\tau|)$. We show that Φ' and Ψ' witness that $P \leq_{sW} Q$.

Take any P-instance C. Since C' is a P-instance, $\Phi'(C) = \Phi(C')$ is a Q-instance. Let s be any Q-solution to $\Phi(C')$. Then $\Psi(C' \oplus s)$ is a P-solution to C'. In particular, $\Psi(C' \oplus s)$ converges. Since C' extends τ, by our construction of τ, we have that $\Psi(\tau \oplus s)\downarrow = \Psi(C' \oplus s)\downarrow$. Hence $\Psi(\tau \oplus s)$ is a P-solution to C'. We conclude that $\Psi'(s) = \Theta(\Psi(\tau \oplus s) \oplus |\tau|)$ is a P-solution to C. □

It is easy to see that RT^1_k (and finite parallel products of RT^1_k) satisfy the properties for P and Q in Lemma 5.1. Therefore

Corollary 5.2. *If* $\prod_{m=0}^n \mathsf{RT}^1_{k_m} \leq_W \mathsf{RT}^1_N$, *then* $\prod_{m=0}^n \mathsf{RT}^1_{k_m} \leq_{sW} \mathsf{RT}^1_N$.

Optimality then follows from a counting argument:

Proposition 5.3. *If* $\prod_{m=0}^n \mathsf{RT}^1_{k_n} \leq_{sW} \mathsf{RT}^1_N$, *then* $N \geq \prod_{m=0}^n k_m$.

Proof. Fix Φ and Ψ witnessing that $\prod_{m=0}^n \mathsf{RT}^1_{k_m} \leq_{sW} \mathsf{RT}^1_N$. We show that for each $(a_0, \ldots, a_n) \in \prod_{m=0}^n k_m$, there is some $c < N$ such that $\Psi(c) = (a_0, \ldots, a_n)$.

Consider the tuple of constant colorings $(a_0^\omega, \ldots, a_n^\omega)$. This is a $\prod_{m=0}^n \mathsf{RT}^1_{k_m}$-instance, so $\Phi(a_0^\omega, \ldots, a_n^\omega)$ is an RT^1_N-instance with some solution c. $\Psi(c)$ must be a solution to $(a_0^\omega, \ldots, a_n^\omega)$, so $\Psi(c) = (a_0, \ldots, a_n)$. □

Corollary 5.4. *If* $\prod_{m=0}^n \mathsf{RT}^1_{k_n} \leq_W \mathsf{RT}^1_N$, *then* $N \geq \prod_{m=0}^n k_m$.

Therefore the right-hand side of $\prod_{m=0}^n \mathsf{RT}^1_{k_m} \leq_{sW} \mathsf{RT}^1_{\prod_{m=0}^n k_m}$ is optimal, with regards to both $\leq_W$ and $\leq_{sW}$.

5.2 How many colors can a product of colorings handle?

In contrast to Corollary 5.4, we will see that $\mathsf{RT}^1_{\prod_{m=0}^n k_m} \not\leq_W \prod_{m=0}^n \mathsf{RT}^1_{k_m}$ for all $n \geq 1$ and $k_0, \ldots, k_n \geq 2$ (Proposition 5.13). In the rest of this section, we attempt to find the smallest N such that

$$\mathsf{RT}^1_N \not\leq_W \prod_{m=0}^{n} \mathsf{RT}^1_{k_m}.$$

We start by giving a lower bound for N:

Proposition 5.5. *For all $n \geq 1$ and $k_0, \dots, k_n \geq 2$,*

$$\mathsf{RT}^1_{1+\sum_{m=0}^{n}(k_m-1)} \leq_{sW} \prod_{m=0}^{n} \mathsf{RT}^1_{k_m}.$$

Proof. Suppose we are given an instance c of $\mathsf{RT}^1_{1+\sum_{m=0}^{n}(k_m-1)}$. For $0 \leq m \leq n$, we define colorings

$$d_m : \mathbb{N} \to \left\{ \sum_{i=0}^{m-1}(k_i - 1), \dots, \sum_{i=0}^{m}(k_i - 1) \right\}$$

as follows. Note that for each m, d_m will be an k_m-coloring.

For each m and x, we define $d_m(x)$ as follows. First check which color among $0, \dots, \sum_{i=0}^{m}(k_i - 1)$ appears most often among $c(0), \dots, c(x)$. (Resolve ties by picking the $<_{\mathbb{N}}$-least color.) If said color is among $0, \dots, \sum_{i=0}^{m-1}(k_i - 1)$, let $d_m(x) = \sum_{i=0}^{m-1}(k_i - 1)$. Otherwise, let $d_m(x)$ be said color.

Now, if for each m, the color a_m appears infinitely often in d_m, we want to compute a color which appears infinitely often in c. Start by considering a_n. If $a_n \neq \sum_{i=0}^{n-1}(k_i - 1)$, then for infinitely many x, a_n appears most often among $c(0), \dots, c(x)$. In particular, a_n appears infinitely often in c.

On the other hand, if $a_n = \sum_{i=0}^{n-1}(k_i - 1)$, then for infinitely many x, some color among $0, \dots, \sum_{i=0}^{n-1}(k_i - 1)$ appears most often among $c(0), \dots, c(x)$. By the pigeonhole principle, some color among $0, \dots, \sum_{i=0}^{n-1}(k_i - 1)$ appears infinitely often in c. We then proceed to consider a_{n-1} and repeat the above case division. Eventually we either reach some a_m which is not equal to $\sum_{i=0}^{m-1}(k_i - 1)$, in which case a_m appears infinitely often in c, or we reach $a_0 = 0$, in which case 0 appears infinitely often in c. $\square$

In order to obtain upper bounds for N, we begin by restricting the reductions that we need to diagonalize against. Firstly, by Lemma 5.1, we need only handle strong Weihrauch reductions:

Proposition 5.6. *If* $\mathsf{RT}^1_N \leq_W \prod_{m=0}^n \mathsf{RT}^1_{k_m}$*, then* $\mathsf{RT}^1_N \leq_{sW} \prod_{m=0}^n \mathsf{RT}^1_{k_m}$*.*

We can impose a further restriction:

Proposition 5.7. *Suppose* $\mathsf{RT}^1_N \leq_{sW} \prod_{m=0}^n \mathsf{RT}^1_{k_m}$ *via some forward functionals* Φ_m*,* $0 \leq m \leq n$*, where* Φ_m *computes the* m^{th} *coloring in the* $\prod_{m=0}^n \mathsf{RT}^1_{k_m}$*-instance, and a backward functional* Ψ*. Then for any* $c < N$*, there exists* $(a_0, \ldots, a_n)$ *where each* $a_m < k_m$ *and* $\Psi(a_0, \ldots, a_n) = c$*.*

Proof. Given $c < N$, consider the coloring C which is constantly c. Then $\Phi_0^C, \ldots, \Phi_n^C$ is a $\prod_{m=0}^n \mathsf{RT}^1_{k_m}$-instance. Hence it has some solution $(a_0, \ldots, a_n)$. The only solution to C is c, so $\Psi(a_0, \ldots, a_n)$ must be c. $\square$

Combining the previous two propositions, we obtain

Corollary 5.8. *Suppose* $\mathsf{RT}^1_N \leq_W \prod_{m=0}^n \mathsf{RT}^1_{k_m}$*. Then* $\mathsf{RT}^1_N \leq_{sW} \prod_{m=0}^n \mathsf{RT}^1_{k_m}$*, as witnessed by some* Φ_m*,* $0 \leq m \leq n$ *and* Ψ *where* $\Psi : \prod_{m=0}^n k_m \to N$ *is a partial surjection.*

Henceforth, we will always assume that our reductions from RT^1_N to $\prod_{m=0}^n \mathsf{RT}^1_{k_m}$ have the above special form. In order to diagonalize against such reductions, it will be convenient to have the following notion of covering a tuple of colors using a set of tuples of colors.

Definition 5.9. If $X \subseteq \prod_{m=0}^n k_m$ and $(c_0, \dots, c_n) \in \prod_{m=0}^n k_m$, we say that X *covers* $(c_0, \dots, c_n)$ if for each $0 \le m \le n$, there is a $(a_0, \dots, a_n) \in X$ such that $a_m = c_m$.

Observe that if C is a $\prod_{m=0}^n \mathsf{RT}^1_{k_m}$-instance whose solution set contains X, and X covers $(c_0, \dots, c_n)$, then $(c_0, \dots, c_n)$ is also a solution to C.

The following terminology will also be useful.

Definition 5.10. If we fix a partial surjection $\Psi : \prod_{m=0}^n k_m \to N$, then we refer to each $\Psi^{-1}(\{c\})$, $c < N$ as a *group*. We call a group of size one a *singleton*.

We now work towards an upper bound ($\approx (\prod k_m)/2$) for N. Suppose we want to show that $\mathsf{RT}^1_N \not\le_W \prod_{m=0}^n \mathsf{RT}^1_{k_m}$ for some N. Towards a contradiction, we may (by Corollary 5.8) fix Φ_m, $0 \le m \le n$ and Ψ witnessing that $\mathsf{RT}^1_N \le_{sW} \prod_m \mathsf{RT}^1_{k_m}$ such that Ψ is a partial surjection from $\prod_{m=0}^n k_m$ to N. We aim to construct $C : \mathbb{N} \to N$ and some $(a_0, \dots, a_n)$ such that $(a_0, \dots, a_n)$ is a solution to $\Phi_0^C, \dots, \Phi_n^C$, yet $\Psi(a_0, \dots, a_n)$ is not a solution to C.

Our basic strategy is to choose N large enough so that the following combinatorial property holds for all partial surjections $\Psi : \prod_{m=0}^n k_m \to N$:

> There is some nonempty $S \subsetneq N$ such that for any set of $(a_0, \dots, a_n)$'s whose image under Ψ is exactly S, the $(a_0, \dots, a_n)$'s cover some $(b_0, \dots, b_n)$ which maps outside S under Ψ. $(*)$

Assuming $(*)$, we may construct C by repeatedly looping through colors in S: for each $c \in S$, extend constantly by c until there is some $(a_0, \dots, a_n)$, which maps to c under Ψ, such that for all $0 \le m \le n$, Φ_m^C has some new element of color a_m.

(This must happen eventually: if C is the RT^1_N instance produced by extending the current finite coloring by c forever, then $\Phi_0^C, \ldots, \Phi_n^C$ is a $\prod_{m=0}^n \mathsf{RT}^1_{k_m}$-instance with some solution $(a_0, \ldots, a_n)$. Then $\Psi(a_0, \ldots, a_n) = c$, and for each $0 \leq m \leq n$, some new element of color a_m must appear at some finite stage of Φ_m^C.)

Then for all $c \in S$, there is some $(a_0, \ldots, a_n)$ such that $\Psi(a_0, \ldots, a_n) = c$ and $(a_0, \ldots, a_n)$ is a solution to $\Phi_0^C, \ldots, \Phi_n^C$. But then the $(a_0, \ldots, a_n)$'s cover some $(b_0, \ldots, b_n)$ which maps outside S under Ψ. It follows that $(b_0, \ldots, b_n)$ is also a solution to $\Phi_0^C, \ldots, \Phi_n^C$. But $\Psi(b_0, \ldots, b_n) \notin S$ and is hence not a solution to C, contradiction. This shows that $\mathsf{RT}^1_N \not\leq_W \prod_{m=0}^n \mathsf{RT}^1_{k_m}$.

The above strategy may be applied as follows:

Proposition 5.11. *If $N > \max\{(k_0 \cdot k_1)/2, k_0 + k_1 - 1\}$, then $\mathsf{RT}^1_N \not\leq_W \mathsf{RT}^1_{k_0} \times \mathsf{RT}^1_{k_1}$.*

Proof. By the previous discussion, it suffices to show that $(*)$ holds. Since $N > (k_0 \cdot k_1)/2$, by a counting argument, Ψ must have at least one singleton (a_0, a_1). Note that there are $1 + (k_0 - 1) + (k_1 - 1) = k_0 + k_1 - 1$ many pairs in $k_0 \times k_1$ which share some color with (a_0, a_1). But $N > k_0 + k_1 - 1$, so there is some group G such that none of its pairs share any colors with (a_0, a_1). In other words, for every pair in G, the set containing it and (a_0, a_1) covers a pair outside G. Let S be the image of (a_0, a_1) and G under Ψ. Then S witnesses that $(*)$ holds. □

Corollary 5.12. *We have that*

$$\begin{aligned}
\mathsf{RT}^1_4 &\not\leq_W \mathsf{RT}^1_2 \times \mathsf{RT}^1_2, & \mathsf{RT}^1_5 &\not\leq_W \mathsf{RT}^1_2 \times \mathsf{RT}^1_3,\\
\mathsf{RT}^1_6 &\not\leq_W \mathsf{RT}^1_2 \times \mathsf{RT}^1_4, & \mathsf{RT}^1_6 &\not\leq_W \mathsf{RT}^1_3 \times \mathsf{RT}^1_3,\\
\mathsf{RT}^1_7 &\not\leq_W \mathsf{RT}^1_2 \times \mathsf{RT}^1_5, & \mathsf{RT}^1_7 &\not\leq_W \mathsf{RT}^1_3 \times \mathsf{RT}^1_4,\\
\mathsf{RT}^1_8 &\not\leq_W \mathsf{RT}^1_2 \times \mathsf{RT}^1_6, & \mathsf{RT}^1_8 &\not\leq_W \mathsf{RT}^1_3 \times \mathsf{RT}^1_5.
\end{aligned}$$

Note that Proposition 5.5 implies that $\mathsf{RT}^1_{k_0+k_1-1} \leq_W \mathsf{RT}^1_{k_0} \times \mathsf{RT}^1_{k_1}$. Hence all of the nonreductions in Corollary 5.12 are sharp.

We can derive more results using variations of the argument in Proposition 5.11.

Proposition 5.13. *If $N > (\max k_m + \prod k_m)/2$, then $\mathsf{RT}^1_N \not\leq_W \prod_{m=0}^n \mathsf{RT}^1_{k_m}$.*

Proof. As before, we show that $(*)$ holds. By a counting argument, Ψ must have at least $1 + \max k_m$ many $(a_0, \ldots, a_n)$ which are singletons. Among these singletons, there must be two of them which differ in at least two entries, i.e., the set consisting of these two singletons cover a new tuple of colors. We can then take S to be the image of two such singletons under Ψ. $\square$

We can improve on this asymptotically, but even then this seems to be far from optimal.

Proposition 5.14. *If $N > \max\{(2 + \prod k_m)/2, \max k_m - 1 + (\prod k_m)/3\}$, then $\mathsf{RT}^1_N \not\leq_W \prod_{m=0}^n \mathsf{RT}^1_{k_m}$.*

Proof. As before, we show that $(*)$ holds. Since $N > (2 + \prod k_m)/2$, Ψ must have at least three singletons.

<u>Case 1.</u> If there are two singletons which differ in at least two entries, then we may take S to be the image of two such singletons under Ψ, as in Proposition 5.13.

<u>Case 2.</u> Otherwise, all of the singletons share exactly one common entry. So there are some $0 \leq m \leq n$ and $3 \leq l \leq k_m$ such that there are exactly l many singletons and all of them are of the form $(a_0, \ldots, a_{m-1}, b, a_{m+1}, \ldots, a_n)$, where $b < k_m$.

We claim that there are at least $k_m + 1$ many groups of size $< l$. If not, by a counting argument, there are at least

$$\begin{aligned}
&1 \cdot l + 2 \cdot (k_m - l) + l \cdot (N - k_m) \\
&= l + 2k_m - 2l + lN - lk_m \\
&> l\left(\max k_m - 1 + \frac{\prod k_m}{3}\right) + 2k_m - l - lk_m \\
&\geq lk_m - l + \prod k_m + 2k_m - l - lk_m \\
&\geq \prod k_m
\end{aligned}$$

many tuples, contradiction.

By the claim, there is a group U of size $< l$ which does not contain any tuple of the form $(a_0, \ldots, a_{m-1}, b, a_{m+1}, \ldots, a_n)$. Since $|U| < l$, there is a singleton $(a_0, \ldots, a_{m-1}, b, a_{m+1}, \ldots, a_n)$ such that b does not appear in any tuple in U. Then for any tuple in U, the set containing it and $(a_0, \ldots, a_{m-1}, b, a_{m+1}, \ldots, a_n)$ covers some tuple outside U, so we can take S to be the image of U and said singleton. $\square$

The lower bound in Proposition 5.5 is, in general, much smaller than the upper bounds in Propositions 5.11, 5.13, and 5.14. Observe that in all of our proofs, the sets S consist of two elements, at least one of which is the image of a singleton under Ψ. However, Ψ may not have any singletons, for example in a hypothetical reduction witnessing that $\mathsf{RT}^1_8 \leq_W \mathsf{RT}^1_4 \times \mathsf{RT}^1_4$. Also, there may not be any S which has exactly two elements and satisfies $(*)$, e.g., consider $\Psi : 4 \times 4 \to 8$ as represented in the grid below. Ψ maps $(i, j) \in 4 \times 4$ to the number in the $(i, j)^{\text{th}}$

position.

$$\begin{array}{cccc} 0 & 3 & 2 & 6 \\ 0 & 4 & 5 & 7 \\ 1 & 2 & 3 & 7 \\ 1 & 4 & 5 & 6 \end{array}$$

One can check that for any $c, d < 8$, there is a point labeled c which shares a row or column with a point labeled d. That means that $S = \{c, d\}$ fails to satisfy $(*)$.

Therefore, new techniques will be required to close the gap between our lower and upper bounds. We conclude this section by giving an ad hoc proof that $\mathsf{RT}^1_8 \not\leq_W \mathsf{RT}^1_4 \times \mathsf{RT}^1_4$, which is the smallest case not resolved by Corollary 5.12. In order to do so, we will show that there exists some S which satisfies $(*)$ and has exactly *three* elements.

Before specializing to the case of $\mathsf{RT}^1_8 \not\leq_W \mathsf{RT}^1_4 \times \mathsf{RT}^1_4$, we consider a more general context: let $k_0, k_1 \geq 2$ and fix a partial surjection $\Psi : k_0 \times k_1 \to N$ (i.e., a potential backward reduction for $\mathsf{RT}^1_N \not\leq_W \mathsf{RT}^1_{k_0} \times \mathsf{RT}^1_{k_1}$.) We say that a collection of three groups is *bad* if its image under Ψ does not satisfy $(*)$. We may characterize the bad collections of three groups:

Lemma 5.15. *Let $k_0, k_1 \geq 2$ and let $\Psi : k_0 \times k_1 \to N$ be a partial surjection. A collection of three groups is bad if and only if their union contains either:*

1. *three pairs in a row/column (e.g., (a, b_0), (a, b_1), (a, b_2)), with one pair from each of the three groups;*
2. *four pairs which form a rectangle (e.g., (a_0, b_0), (a_0, b_1), (a_1, b_0), (a_1, b_1)), with at least one pair from each of the three groups.*

Proof. $(\Leftarrow)$. If (1) holds, the three pairs in question do not cover any new pair. If

(2) holds, pick three out of the four pairs such that one pair from each of the three groups is picked. Then these three pairs cover exactly one other pair (the fourth). But the fourth pair is already contained in the union of the three groups.

($\Rightarrow$). Suppose that we have a bad collection of three groups. Without loss of generality, we may pick one pair (a_i, b_i) from each group such that the three pairs (a_0, b_0), (a_1, b_1), and (a_2, b_2) witness badness.

<u>Case 1.</u> (a_0, b_0), (a_1, b_1), and (a_2, b_2) lie in the same row or column. Then they satisfy (1).

<u>Case 2.</u> Two out of the three pairs, say (a_0, b_0) and (a_1, b_1), lie in the same row or column (i.e., $a_0 = a_1$ or $b_0 = b_1$). Without loss of generality, suppose that $b_0 = b_1$. Note that (a_0, b_0), (a_0, b_1), and (a_2, b_2) cover (a_2, b_0), (a_0, b_2), and (a_2, b_1). Therefore by badness, the latter three pairs lie in the union of the three groups.

If (a_0, b_0), (a_1, b_1), and (a_2, b_2) are vertices of a rectangle (i.e., $b_2 = b_0$ or $b_2 = b_1$), then we satisfy (2). Otherwise, we consider cases depending on which group contains (a_2, b_0). In all cases, we satisfy either (1) or (2). See Figure 5.1 for an illustration.

<u>Case 3.</u> None of the three pairs lie in the same row or column. Note that by badness, (a_0, b_1), (a_1, b_0), (a_0, b_2), (a_2, b_0), (a_1, b_2), and (a_2, b_1) all lie in the union of the three groups. We consider cases depending on which group contains (a_2, b_1). See Figure 5.2 for an illustration.

<u>Case 3a.</u> (a_2, b_1) and (a_0, b_0) lie in the same group. Then we satisfy (2): (a_1, b_2), (a_1, b_1), (a_2, b_1), and (a_2, b_2) form a rectangle with at least one pair from each of the three groups.

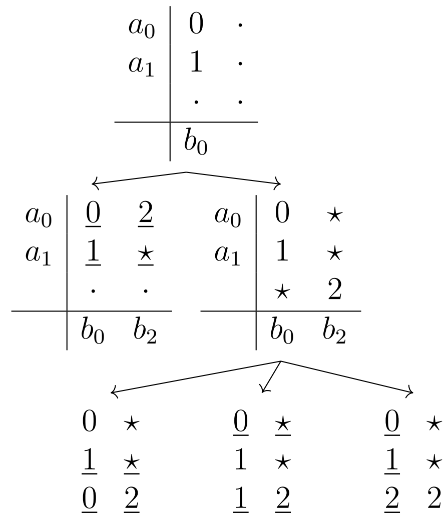

Figure 5.1: Case 2 in Lemma 5.15, assuming that $b_0 = b_1$. In the array on the top level, 0 lies in position (a_0, b_0) and 1 lies in position (a_1, b_0), meaning that $\Psi(a_0, b_0) = 0$ and $\Psi(a_1, b_0) = 1$. We have yet to label position (a_2, b_2). The middle level represents cases depending on whether a_2 equals some a_i, or not. If a star lies in position (a, b), then (a, b) is known (by badness) to lie in the union of the bad collection of three groups. Sets of pairs that satisfy (1) or (2) are underlined. The bottom level represents cases depending on which of the three groups contains (a_2, b_0). For example, in the array on the bottom right, 2 lies in positions (a_2, b_0) and (a_2, b_2), meaning that $\Psi(a_2, b_0) = \Psi(a_2, b_2) = 2$ and hence (a_2, b_0) and (a_2, b_2) lie in the same group. Then (a_0, b_0), (a_1, b_0), and (a_2, b_0) lie in a column, satisfying (1).

Case 3b. (a_2, b_1) and (a_1, b_1) lie in the same group. Then we consider cases depending on which group contains (a_2, b_0). In all cases, we satisfy either (1) or (2).

Case 3c. (a_2, b_1) and (a_2, b_2) lie in the same group. We consider cases depending on which group contains (a_0, b_1). The argument is symmetric to Case 3b. $\square$

Proposition 5.16. $\mathsf{RT}^1_8 \not\leq_W \mathsf{RT}^1_4 \times \mathsf{RT}^1_4$.

Proof. Towards a contradiction, fix forward functionals Φ_0, Φ_1 and a partial surjection $\Psi : 4 \times 4 \to 8$ witnessing that $\mathsf{RT}^1_8 \leq_W \mathsf{RT}^1_4 \times \mathsf{RT}^1_4$. If Ψ has any singletons, we can derive a contradiction using the proof of Proposition 5.11. Hence we assume

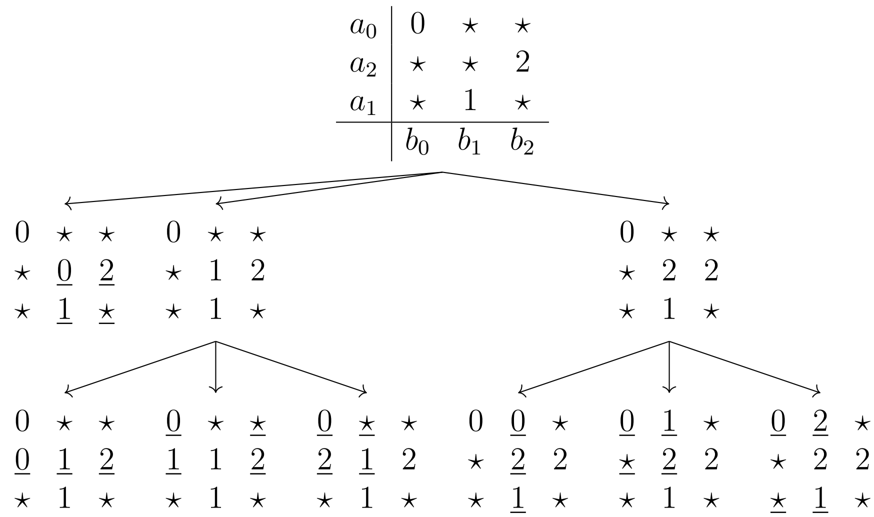

Figure 5.2: Case 3 in Lemma 5.15. In the array on the top level, for each $i < 3$, i lies in position (a_i, b_i), meaning that $\Psi(a_i, b_i) = i$. On the middle level, we have Case 3a on the left, followed by Case 3b and 3c. On the bottom level, we have various subcases. For example, in the array on the bottom right, 0 lies in position (a_0, b_0), 2 lies in position (a_0, b_1), and 1 lies in position (a_1, b_1). Together with (a_1, b_0), they form a rectangle satisfying (2).

that Ψ has no singletons. There are sixteen pairs in 4×4, so Ψ must be total, and all of the eight groups in Ψ must contain exactly two pairs each.

As discussed previously, we derive a contradiction by producing a set S which satisfies $(*)$ and consists of three elements. In other words, we show that there is a collection of three groups which is not bad. To that end, we give an upper bound for the number of bad collections of three groups. Since each group contains exactly two pairs, it is either contained in a row or column, or lies in diagonal position. Let k be the number of groups which are contained in some row or column.

First, we give an upper bound for the number of collections which satisfy (2) in Lemma 5.15. It suffices to give an upper bound for the number of rectangles which intersect at most three groups. Such rectangles have two possible forms, and we count those cases separately.

Case 1. The rectangle contains at least one of those k groups. There are at most $(4-1)k = 3k$ many such rectangles.

Case 2. The rectangle contains at least one group in diagonal position. There are at most $8-k$ many such rectangles.

Therefore, there are at most $3k + (8-k) = 2k+8$ many rectangles which intersect at most three groups. So there are at most $2k+8$ many collections which satisfy (2).

Next, we give an upper bound for the number of collections which satisfy (1) in Lemma 5.15.

Case 1. If a row/column contains two groups (and hence nothing else), then said row/column does not contribute to our upper bound. Let l be the number of such rows and columns. Note that $2l \leq k$.

Case 2. If a row/column contains one group, as well as two other vertices from two different groups, then said row/column contributes one collection to our upper bound. There are $k-2l$ many such rows/columns.

Case 3. Finally, the remaining $8+l-k$ many rows or columns contribute $\binom{4}{3} = 4$ collections each.

Therefore, there are at most

$$l \cdot 0 + (k-2l) \cdot 1 + (8+l-k) \cdot 4 = 32 - 3k + 2l \leq 32 - 2k$$

many collections which satisfy (1).

We conclude that there are at most $(2k+8) + (32-2k) = 40$ bad collections of three groups. There are $\binom{8}{3} = 56 > 40$ collections of three groups in total, so we can define S to be the image under Ψ of any collection which is not bad. Then S satisfies $(*)$, contradiction. $\square$

CHAPTER 6

A Σ^1_1 AXIOM OF FINITE CHOICE

In this chapter, all theories of second-order arithmetic implicitly contain RCA_0.

6.1 Theories of hyperarithmetic analysis

Van Wesep [44, Section 2] showed in his thesis that given any theory of second-order arithmetic all of whose ω-models are *hyp closed* (i.e., closed under join and hyperarithmetic reduction $\leq_h$), there exists a strictly weaker such theory. In particular, there is no theory of second-order arithmetic whose ω-models are exactly the hyp closed models. We present his proof below, with minor simplifications.

Consider the theory S consisting of, for each $e \in W$, the sentence "for all Z, there is a jump hierarchy along the e^{th} computable well-ordering starting with Z". Observe that every ω-model which is hyp closed satisfies S. But we overshoot:

Proposition 6.1. *There is an ω-model M of S which is not hyp closed.*

Proof. Fix any Z such that $\omega_1^Z > \omega_1^{CK}$ (equivalently, $Z \geq_h W$). Define M to be the class of all X which are computable in some jump hierarchy along a computable (not Z-computable!) well-ordering starting with Z. □

Theorem 6.2 (van Wesep)**.** *Let T be a theory of second-order arithmetic all of whose ω-models are hyp closed. Then there is some ω-model M which is hyp closed, yet does not satisfy T.*

Proof. By the previous proposition, there is an ω-model of S which does not satisfy T. Fix $\varphi \in \mathsf{T}$ such that $\mathsf{S} + \neg\varphi$ has an ω-model.

Now, S (and hence $\mathsf{S} + \neg\varphi$) is a Π^1_1 set of sentences. Hence being an ω-model of $\mathsf{S} + \neg\varphi$ is Σ^1_1. By Gandy's basis theorem, $\mathsf{S} + \neg\varphi$ has an ω-model $M <_h W$.

We show that M is hyp closed: note that for every $Z \in M$, $Z <_h W$, so $\omega_1^Z = \omega_1^{CK}$. Since M satisfies S, it follows that M is hyp closed as desired. □

In fact:

Theorem 6.3 (van Wesep)**.** *For any* T *all of whose* ω*-models are hyp closed, there is some* T' *with the same property but with more* ω*-models than* T*.*

Proof. By the proof of the previous theorem, fix $M <_h W$ which is hyp closed, and is a model of S but not T. We aim to construct T', such that all ω-models of T' are hyp closed, and T' has more ω-models than T as witnessed by M. It suffices to construct a sentence ρ such that $M \models \rho$ and every model of ρ is hyp closed. Then $\mathsf{T}' = \{\psi \vee \rho : \psi \in \mathsf{T}\}$ would satisfy the desired properties.

Since $W \not\leq_h M$, by overspill, there must be an ill-founded computable linear ordering L_0 such that M satisfies the following sentence ρ:

$$L_0 \text{ is well-founded} \wedge \forall Z(\text{there is a jump hierarchy along } \omega \cdot L_0 \text{ starting with } Z).$$

We show that every ω-model N of ρ is hyp closed. We know that for any $Z \in N$, N contains some jump hierarchy Y along $\omega \cdot L_0$ starting with Z. Since $\omega \cdot L_0$ is ill-founded, Y computes a jump hierarchy starting with Z on every computable well-ordering (see Sacks [39, III.3.3].) So it remains to show that for any $Z \in N$, $\omega_1^Z = \omega_1^{CK}$.

Suppose not, i.e., there is some $Z \in N$ and some Z-computable well-ordering L_1 of length ω_1^{CK}. Now, let $Y \in N$ be a jump hierarchy along $\omega \cdot L_0$ starting with Z. Observe that Y computes a comparison map between L_0 and L_1.

Since there is no embedding from L_0 into L_1 (L_0 is ill-founded while L_1 is well-founded), this map must be an isomorphism between L_1 and an initial segment of L_0. But the well-founded part of L_0 has ordertype ω_1^{CK} (see Sacks [39, III.2.2(i)]), so this map (together with Z) computes a proper cut in L_0.

Hence N contains a proper cut in L_0. But then N sees that L_0 is ill-founded, contradicting our assumption that N satisfies ρ. □

Even though there is no theory whose ω-models are exactly the hyp closed ones, there are several theories that come close, in the following sense:

Definition 6.4. A theory T is a *theory of hyperarithmetic analysis* if:

- every ω-model of T is hyp closed;
- for every $Y \subseteq \mathbb{N}$, $\mathrm{HYP}(Y)$ is a model of T.

We say that φ is a *theorem of hyperarithmetic analysis* if $\mathsf{RCA}_0 + \varphi$ is a theory of hyperarithmetic analysis.

Note that by Theorem 6.3, there is no weakest theory of hyperarithmetic analysis.

Of particular interest to us are the following theories:

Definition 6.5. The system of Σ^1_1 *axiom of choice*, denoted Σ^1_1-AC_0, consists of the axiom schema

$$\forall n \exists Y \varphi(n, Y) \to \exists (Z_n)_n \forall n \varphi(n, Z_n),$$

for any $\varphi(n, Y)$ which is arithmetic.

The system of Δ^1_1 *comprehension*, denoted $\Delta^1_1\text{-}\mathsf{CA}_0$, consists of the axiom schema

$$\forall n(\varphi(n) \leftrightarrow \neg\psi(n)) \to \exists X \forall n(n \in X \leftrightarrow \varphi(n)),$$

for any $\varphi(n)$ and $\psi(n)$ which are Σ^1_1.

The system of Σ^1_1 *axiom of unique choice*, denoted *unique*-$\Sigma^1_1\text{-}\mathsf{AC}_0$, consists of the axiom schema

$$\forall n(\exists \text{ unique } Y)\varphi(n, Y) \to \exists (Z_n)_n \forall n \varphi(n, Z_n),$$

for any $\varphi(n, Y)$ which is arithmetic.

Note that even though HYP satisfies $\Sigma^1_1\text{-}\mathsf{AC}_0$, that does not mean that if $\varphi(n, Y)$ is arithmetic, then for all n, there must be some hyperarithmetical Y such that $\varphi(n, Y)$ holds. In fact, Kleene showed that the latter statement is false; see [39, II.1.4]. Rather, if there is n such that $\varphi(n, \cdot)$ has no hyperarithmetical solution, then φ is not an instance of $\Sigma^1_1\text{-}\mathsf{AC}_0$ in HYP.

What we call unique-$\Sigma^1_1\text{-}\mathsf{AC}_0$ is sometimes known as $\Pi^0_{(\omega)}$-replacement ([44]), Π^1_0-replacement ([43]), or weak-$\Sigma^1_1\text{-}\mathsf{AC}_0$ ([42, VIII.4.12]). Our choice of nomenclature should be compared with the following new variant of $\Sigma^1_1\text{-}\mathsf{AC}_0$:

Definition 6.6. The system of Σ^1_1 *axiom of finite choice*, denoted *finite*-$\Sigma^1_1\text{-}\mathsf{AC}_0$, consists of the axiom schema

$$\forall n(\exists \text{ finitely many } Y)\varphi(n, Y) \to \exists (Z_n)_n \forall n \varphi(n, Z_n),$$

for any $\varphi(n, Y)$ which is arithmetic.

Note that by a lemma of Simpson [42, V.5.4], finite-$\Sigma^1_1\text{-}\mathsf{AC}_0$ is equivalent to the following statement: "if $(T_n)_n$ is a sequence of subtrees of $\mathbb{N}^{<\mathbb{N}}$, each of which has finitely many paths, then there is a sequence $(X_n)_n$ such that for each n, X_n is a path on T_n".

Trivially, finite-Σ^1_1-AC_0 lies between unique-Σ^1_1-AC_0 and Σ^1_1-AC_0. Hence it is a theory of hyperarithmetic analysis. In this chapter, we explore the relationships between finite-Σ^1_1-AC_0 and other known theorems of hyperarithmetic analysis.

6.2 Arithmetic Bolzano-Weierstrass implies finite-Σ^1_1-AC_0

The arithmetic Bolzano-Weierstrass theorem was introduced by Friedman [18]. Our definition follows Conidis [13].

Definition 6.7. The *arithmetic Bolzano-Weierstrass theorem*, denoted ABW_0, states that if $A(X)$ is an arithmetic predicate on $2^{\mathbb{N}}$, then either $A(X)$ has finitely many solutions, or the set of A-solutions $\{X \in 2^{\mathbb{N}} : A(X)\}$ has an accumulation point.

Friedman asserted that ABW_0 follows from Σ^1_1-AC_0. Conidis [13, Theorem 2.1] furnished a proof of that statement. In addition, Conidis showed that ABW_0, together with the induction schema for Σ^1_1 formulas (denoted $I\Sigma^1_1$), implies unique-Σ^1_1-AC_0. We adapt his proof to show that:

Proposition 6.8 ($I\Sigma^1_1$)**.** ABW_0 *implies finite-*Σ^1_1-AC_0.

Proof. Suppose that $A(n, Y)$ is an instance of finite-Σ^1_1-AC_0, i.e., for each n, $A(n, Y)$ has finitely many solutions. Without loss of generality, we may assume that $A(n, \emptyset)$ always fails.

Define $B((X_n)_n)$ to hold if

$$\exists n_0[(\forall n \leq n_0)A(n, X_n) \wedge (\forall n > n_0)[X_n = \emptyset]].$$

Observe that $B((X_n)_n)$ is an arithmetic predicate on $2^{\mathbb{N}}$. Using $I\Sigma^1_1$ and the assumption that for each n, $A(n, Y)$ has a solution, we can show that for each n_0, $B((X_n)_n)$ has at least n_0 distinct solutions. (Since $A(n, \emptyset)$ always fails, the solutions we construct are distinct.) Hence $B((X_n)_n)$ is an instance of ABW.

Hence we may apply ABW to B to obtain an accumulation point $(Y_n)_n$ of $\{(X_n)_n : B((X_n)_n)\}$. We claim that for all n, $A(n, Y_n)$ holds.

Suppose towards a contradiction that $A(k, Y_k)$ fails. Since $A(k, \cdot)$ has only finitely many solutions, there is sufficiently large m such that $Y_k \upharpoonright m \neq Y \upharpoonright m$ for every Y such that $A(k, Y)$ holds.

Now, by our choice of $(Y_n)_n$, there are infinitely many $(X_n)_n$ satisfying B such that X_k extends $Y_k \upharpoonright m$. For any such $(X_n)_n$, $A(k, X_k)$ fails, so by definition of B, $X_n = \emptyset$ for all $n \geq k$.

But for each $n < k$, $A(n, \cdot)$ has at most finitely many solutions, so there cannot be infinitely many $(X_n)_n$ satisfying the above conditions. Contradiction. We have showed that $(Y_n)_n$ is a finite-Σ^1_1-AC_0 solution to $A(n, Y)$. □

Conidis [13, Theorem 4.1] also showed that ABW_0 does not imply the following theorem, known as INDEC_0:

Theorem 6.9 (Jullien, see Montalbán [30])**.** *Every scattered indecomposable linear ordering is either indecomposable to the right, or indecomposable to the left.*

Montalbán [30] initiated the study of the reverse mathematics of Jullien's theorem. He showed that it is a theorem of hyperarithmetic analysis, making it the first theorem in the literature which is not formulated using concepts from logic and is known to be a theorem of hyperarithmetic analysis. Montalbán also showed that Δ^1_1-CA_0 implies INDEC_0. (Later, Neeman [32], [33] showed that $\mathsf{INDEC}_0 + I\Sigma^1_1$ implies unique-Σ^1_1-AC_0, and $\mathsf{INDEC}_0 + I\Delta^1_1$ does *not* imply unique-Σ^1_1-AC_0.)

It follows from the above results of Conidis and Montalbán that:

Corollary 6.10. *Finite-Σ^1_1-AC_0 does not imply INDEC_0. In particular, finite-Σ^1_1-AC_0 does not imply Δ^1_1-CA_0 or Σ^1_1-AC_0.*

In the next section, we separate finite-Σ^1_1-AC_0 from unique-Σ^1_1-AC_0. In fact, we will show that

Theorem 6.11. *There is an ω-model satisfying Δ^1_1-CA but not finite-Σ^1_1-AC. Hence Δ^1_1-CA_0 does not imply finite-Σ^1_1-AC_0. In particular, unique-Σ^1_1-AC_0 does not imply finite-Σ^1_1-AC_0.*

This result strengthens Conidis [13, Theorem 3.1], which shows that Δ^1_1-CA_0 does not imply ABW_0. We do not know whether our result is strictly stronger than his, i.e., we do not know whether finite-Σ^1_1-AC_0 implies ABW_0.

6.3 Δ^1_1-CA_0 does not imply finite-Σ^1_1-AC_0

In this section, a tree is a prefix-closed subset of $\omega^{<\omega}$. The empty string is denoted by $\emptyset$. If σ is a prefix of τ, we write $\sigma \subseteq \tau$. If σ is a nonempty string, σ^- denotes the prefix of σ of length $|\sigma| - 1$. If σ is a string and T is a tree, $\sigma \cap T$ denotes the longest prefix of σ which lies in T.

6.3.1 The model

We construct an ω-model $M_\infty \subset \mathcal{P}(\omega)$ which satisfies Δ^1_1-CA but not finite-Σ^1_1-AC. To define M_∞ we will construct a generic object

$$\langle T^G, \{\alpha_i^G : i \in \omega\}, h^G\rangle$$

where T^G is a subtree of $\omega^{<\omega}$, each α_i^G is a path on T^G, and $h^G : T^G \to \omega_1^{CK} \cup \{\infty\}$ is the well-founded rank function on T^G, i.e., for all $\sigma \in T^G$, $h^G(\sigma) = \sup\{h^G(\tau)+1 : \tau \in T^G, \tau \supsetneq \sigma\}$ (our convention is that $\infty = \infty + 1$ and $\infty > \infty > \alpha$.)

Then, for each finite $F \subseteq \omega$ (written $F \subset_f \omega$), we define

$$M_F = \{X \subseteq \omega : \exists \mu < \omega_1^{CK}(X \leq_T (T^G \oplus \langle \alpha_i^G\rangle_{i\in F})^{(\mu)})\}$$
$$M_\infty = \bigcup_{F \subset_f \omega} M_F.$$

We will show in Corollary 6.31 that $M_F = \mathrm{HYP}(T^G \oplus \langle\alpha_i^G\rangle_{i\in F})$. Notice that h^G does not appear in the definition of M_∞. Nonetheless it will play a crucial role in showing that M_∞ has the properties we desire.

We briefly sketch the reason why finite-Σ^1_1-AC fails in M_∞. First, we will show in Lemma 6.21 that for each $F \subset_f \omega$, $M_F \cap [T^G] = \{\alpha_i^G : i \in F\}$. This implies that the paths on T^G in M_∞ are exactly $\{\alpha_i^G : i \in \omega\}$. This also implies that M_∞ does not contain any infinite sequence of distinct α_i^G.

Now, for each n, let T_n be the subtree of T^G passing through $\langle n\rangle$. We will use locks in our forcing to ensure that each $[T_n]$ contains finitely many α_i^G. Hence M_∞ thinks that $\langle T_n\rangle_n$ is an instance of finite-Σ^1_1-AC. But the results in the previous paragraph imply that this instance fails to have a solution in M_∞.

For later purposes, we give every element in M_∞ a name. Define the following names by recursion along ω_1^{CK}:

$$H_{1,F} = T^G \oplus \langle \alpha_i^G \rangle_{i \in F}, \quad S_{\mu,F,e} = W_e^{H_{\mu,F}}, \quad H_{\mu,F} = \bigoplus_{\nu<\mu, e\in\omega} S_{\nu,F,e}$$

for $\mu < \omega_1^{CK}$, $F \subset_f \omega$, $e \in \omega$.

6.3.2 The forcing language

We consider a ramified language $\mathcal{L}_\infty$, which extends the language of second-order arithmetic with constants for each element of M_∞, and various types of restricted set variables.

For each $F \subset_f \omega$, the language $\mathcal{L}_F$ is generated by the language of second-order arithmetic, except that the set variables are as follows: for each $D \subseteq F$, there are unranked set variables of the form X_D and ranked set variables of the form X_D^λ for each $\lambda < \omega_1^{CK}$. $\mathcal{L}_F$ also consists of a class C_F of constants which are intended to name every element of M_F:

- $\mathbf{T}$, $\boldsymbol{\alpha}_i$ for $i \in F$;
- for each $\nu < \omega_1^{CK}$, $\mathbf{H}_{\nu,F}$ and $\mathbf{S}_{\nu,F,e}$ for each $e \in \omega$.

If $\mathbf{S}$ is of the form $\mathbf{S}_{\nu,F,e}$, we define $\mathrm{dom}(\mathbf{S})$ to be F. We define C^μ to be set of all constants of the form $\mathbf{H}_{\nu,F}$ or $\mathbf{S}_{\nu,F,e}$ for some $\nu < \mu$.

The language $\mathcal{L}_\infty$ consists of $\bigcup_{F \subset_f \omega} \mathcal{L}_F$, unranked set variables of the form X, and ranked set variables of the form X^λ for each $\lambda < \omega_1^{CK}$.

A variable of the form X_H^ν or X_D is F*-restricted* if $D \subseteq F$. A formula of $\mathcal{L}_\infty$ is F*-restricted* if all of its bounded variables are F-restricted. A formula of $\mathcal{L}_\infty$ is *ranked* if all of its bounded variables are ranked. Every ranked formula ψ of $\mathcal{L}_\infty$ can be assigned a rank below ω_1^{CK}, as follows:

$$\mathrm{rk}(\psi) = \omega^2 \cdot o(\psi) + \omega \cdot r(\psi) + n(\psi),$$

where:

- $o(\psi)$ denotes the least upper bound of

$$\{\nu : \nu \text{ is the superscript of a quantified variable in } \psi\}$$
$$\cup \{\nu + 1 : \text{ some constant of the form } \mathbf{S}_{\nu,F,e} \text{ or } \mathbf{H}_{\nu,F} \text{ occurs in } \psi\};$$

- $r(\psi)$ denotes the number of ranked set quantifiers in ψ;
- $n(\psi)$ denotes the number of connectives.

A formula is Σ-*over*-$\mathcal{L}_F$ if it is built up from ranked F-restricted formulas using $\wedge$, $\forall n$, and $\exists X$. Every Σ^1_1 formula is Σ-over-$\mathcal{L}_F$ for some finite F.

For any formula ψ and any $\mu < \omega_1^{CK}$, we define ψ^μ by replacing every unranked set variable in ψ with its ranked counterpart, i.e., X is replaced by X^μ and X_F is replaced by X_F^μ. Observe that if ψ is Σ-over-$\mathcal{L}_F$ and every constant symbol in ψ lies in C^μ, then $M_\infty \models \psi^\mu \rightarrow \psi$.

We give our formulas Gödel numberings in the usual way. In particular, we fix a computable linear ordering whose well-founded part has ordertype ω_1^{CK}, known as a pseudo-well-ordering (see Harrison [22]). We identify each $\alpha < \omega_1^{CK}$ with its corresponding natural number in the well-founded part.

6.3.3 The forcing notion

The forcing $\mathbb{P}$ consists of tuples $p = \langle T^p, f^p, h^p, l^p \rangle$ where:

(1) $T^p \subseteq \omega^{<\omega}$ is a finite tree;

(2) f^p is a finite partial function from ω to $T^p \setminus \{\emptyset\}$;

(3) $h^p : T^p \to \omega_1^{CK} \cup \{\infty\}$ satisfies the following:

   (a) h^p is a *rank function*, i.e., if $\tau \subsetneq \sigma$, then $h^p(\tau) > h^p(\sigma)$ (our convention is that $\infty > \infty > \alpha$);

   (b) for all $i \in \mathrm{dom}(f^p)$, $h^p(f^p(i)) = \infty$;

   (c) $h^p(\emptyset) = \infty$ and $h^p(\langle n \rangle) = \infty$ for all $\langle n \rangle \in T^p$;

(4) $l^p \subseteq \{n : \langle n \rangle \in T^p\}$. $\langle n \rangle$ is *locked* if $n \in l^p$, and *unlocked* otherwise.

We say that q extends p, written $q \leq p$, if

(5) $T^q \supseteq T^p$;

(6a) for all $i \in \mathrm{dom}(f^p)$, $f^p(i) = f^q(i) \cap T^p$ (old paths cannot be extended in the old tree);

(6b) for all $i \in \mathrm{dom}(f^q) \setminus \mathrm{dom}(f^p)$, $|f^q(i) \cap T^p| \leq 1$ (new paths can only intersect the old tree at the root or at level one);

(7) $h^q \supseteq h^p$;

(8) $l^q \supseteq l^p$ (locked nodes stay locked);

(9) for all $i \in \mathrm{dom}(f^q) \setminus \mathrm{dom}(f^p)$,

$$f^q(i)(0) = n \wedge \exists j [f^p(j)(0) = n] \to n \notin l^p$$

(if a node is locked and there is already a path passing through it, then one cannot add new paths which pass through it.)

Conditions (6a) and (6b) are needed for us to control the complexity of the forcing relation for ranked formulas and Σ-over-$\mathcal{L}_F$ formulas respectively.

We show that the above forcing is transitive.

Proof. Suppose that $r \leq q$ and $q \leq p$. The only nontrivial facts to verify for $r \leq p$ are (6b) and (9). For (6b), we have two cases. If $i \in \mathrm{dom}(f^r)\backslash\mathrm{dom}(f^q)$, then $|f^r(i) \cap T^p| \leq |f^r(i) \cap T^q| \leq 1$ by (6b) for $r \leq q$, as desired. On the other hand, if $i \in \mathrm{dom}(f^q)\backslash\mathrm{dom}(f^p)$, $f^r(i) \cap T^p = (f^r(i) \cap T^q) \cap T^p = f^q(i) \cap T^p$ by (6a) for $r \leq q$. But $|f^q(i) \cap T^p| \leq 1$ by (6b) for $q \leq p$ so $|f^r(i) \cap T^p| \leq 1$ as desired.

As for (9), suppose that $f^r(i)(0) = n$ and there is j such that $f^p(j)(0) = n$. By (6a) for $q \leq p$, $f^q(j)(0) = n$. By (9) for $r \leq q$, $n \notin l^q$. By (8) for $q \leq p$, $l^q \supseteq l^p$. Hence $n \notin l^p$. □

We will take G to be a sufficiently $\mathbb{P}$-generic filter (specifically, G decides all Σ-over-$\mathcal{L}_F$ formulas). Then we may define $T^G = \bigcup_{p \in G} T^p$, $\alpha_i^G = \bigcup_{p \in G} f^p(i)$ for $i \in \omega$, and $h^G = \bigcup_{p \in G} h^p$. By genericity, h^G is the well-founded rank function on T^G.

Just as we did for formulas, we identify each $\alpha < \omega_1^{CK}$ with its corresponding element in the fixed pseudo-well-ordering. When we write $\alpha < \beta$, we always refer to their order as ordinals rather than the natural number ordering. In this way, we can encode $\mathbb{P}$ as a Π^1_1 subset of ω. For each $\alpha < \omega_1^{CK}$, define $\mathbb{P}_\alpha$ to be the set of all conditions p such that the range of h^p is contained in $\alpha \cup \{\infty\}$. Observe that $\mathbb{P} = \bigcup_{\alpha < \omega_1^{CK}} \mathbb{P}_\alpha$ and that the $\mathbb{P}_\alpha$'s are uniformly computable from α.

6.3.4 The forcing relation

The forcing relation for formulas of $\mathcal{L}_\infty$ is defined by recursion as follows:

1. for quantifier-free formulas of arithmetic ψ, $p \Vdash \psi$ if and only if ψ is true;
2. $p \Vdash \boldsymbol{\sigma} \in \mathbf{T}$ if either $|\sigma| < 2$ and $\sigma \in T^p$, or $\sigma^- \in T^p$ and $h^p(\sigma^-) \geq 1$;
3. $p \Vdash \langle \mathbf{n}, \mathbf{m} \rangle \in \boldsymbol{\alpha}_i$ if $i \in \mathrm{dom}(f^p)$ and $f^p(i)(n) = m$;
4. $p \Vdash \langle 0, \sigma \rangle \in \mathbf{H}_{1,F}$ if $p \Vdash \boldsymbol{\sigma} \in T$, and $p \Vdash \langle 1, \langle \mathbf{i}, \langle \mathbf{n}, \mathbf{m} \rangle \rangle \rangle \in \mathbf{H}_{1,F}$ if $i \in F$ and $p \Vdash \langle \mathbf{n}, \mathbf{m} \rangle \in \boldsymbol{\alpha}_i$;
5. for $\nu > 1$, $p \Vdash \mathbf{n} \in \mathbf{S}_{\nu,F,e}$ if $p \Vdash \exists s R(\mathbf{H}_{\nu,F}; \mathbf{e}, s, \mathbf{n})$ where R codes a universal Turing machine;
6. for $\nu > 1$, $p \Vdash \langle \mathbf{e}, \mathbf{n}, \mu \rangle \in \mathbf{H}_{\nu,F}$ if $\mu < \nu$ and $p \Vdash \mathbf{n} \in \mathbf{S}_{\mu,F,e}$;
7. $p \Vdash \forall x \psi(x)$ if for all $n \in \omega$, $p \Vdash \psi(\mathbf{n})$;
8. $p \Vdash \forall X^\lambda_F \psi(X^\lambda_F)$ if for all $\nu < \lambda$, $e \in \omega$, $p \Vdash \psi(\mathbf{S}_{\nu,F,e})$;
9. $p \Vdash \forall X^\lambda \psi(X^\lambda)$ if for all $\nu < \lambda$, $e \in \omega$, $F \subset_f \omega$, $p \Vdash \psi(\mathbf{S}_{\nu,F,e})$;
10. $p \Vdash \forall X_F \psi(X_F)$ if for all $\nu < \omega_1^{CK}$, $e \in \omega$, $p \Vdash \psi(\mathbf{S}_{\nu,F,e})$;
11. $p \Vdash \forall X \psi(X)$ if for all $\nu < \omega_1^{CK}$, $e \in \omega$, $F \subset_f \omega$, $p \Vdash \psi(\mathbf{S}_{\nu,F,e})$;
12. $p \Vdash \varphi \wedge \psi$ if $p \Vdash \varphi$ and $p \Vdash \psi$;
13. $p \Vdash \neg\psi$ if for every $q \leq p$, $q \nVdash \psi$.

Our definitions of M_∞ and the forcing relation are set up so that for sufficiently $\mathbb{P}$-generic G, M_∞ (as defined from G) satisfies ψ if and only if there is $p \in G$ such that $p \Vdash \psi$. A similar statement holds for M_F and ψ which is F-restricted.

6.3.5 Analyzing the forcing relation for ranked formulas

When constructing conditions, we will use the following basic fact about rank functions:

Proposition 6.12. *Let $h : T \to \beta \cup \{\infty\}$ be a rank function. For each $\alpha < \beta$, define the subtree $Q_\alpha = \{\rho \in T : h(\rho) \geq \alpha\}$. Then for every $\tau \in T$ with $h(\tau) \geq \alpha$, we have*

$$h(\tau) \geq \alpha + |\tau|_{Q_\alpha}.$$

We recall the basic notion of retagging used by Steel [43].

Definition 6.13 ([43, Definition 4])**.** If p and p^* are conditions, we say that they are *μ-F-absolute retaggings* if

- $T^p = T^{p^*}$ and $f^p \upharpoonright F = f^{p^*} \upharpoonright F$;
- h^p and h^{p^*} agree on labels $< \mu$;
- if $h^p(\sigma) \geq \mu$, then $h^{p^*}(\sigma) \geq \mu$.

We make some observations:

- μ-F-absolute retagging is an equivalence relation.
- If p and p^* are μ-F-absolute retaggings, then for any $\mu' < \mu$ and any $F' \subseteq F$, p and p^* are μ'-F'-absolute retaggings as well.
- μ-F-absolute retagging is independent of the locks. We will see in Lemma 6.18 that the locks do not affect whether a condition forces a ranked formula. When we analyze the forcing relation for Σ-over-$\mathcal{L}_F$ formulas, we will define two new notions of retagging which do depend on the locks.

We begin by proving a basic retagging lemma, which is a cornerstone of the method of Steel forcing. The presence of locks in our forcing necessitates the assumption that $F \subseteq \mathrm{dom}(f^p)$. This assumption is also made in Conidis [13, Lemma 3.11], but not in Steel [43] or Montalbán [31].

Lemma 6.14. *Let p and p^* be $\omega\beta$-F-absolute retaggings such that $F \subseteq \mathrm{dom}(f^p)$ (hence $F \subseteq \mathrm{dom}(f^{p^*})$ as well.) Then for all $q \leq p$ and all $\gamma < \beta$, there exists $q^* \leq p^*$ such that q and q^* are $\omega\gamma$-F-absolute retaggings.*

Before we prove the lemma, we make a few remarks.

Remark 6.15. The space between $\omega\beta$ and $\omega\gamma$ is so that if, say, $h^{p^*}(\sigma) = \omega\beta$, then the nodes τ extending σ such that $h^q(\tau) \geq \omega\gamma$ can be retagged with $\omega\gamma + n$ for some $n \in \omega$. (Since $\omega\gamma + n < \omega\beta$ for all n, this can be done in a way that makes h^{q^*} a rank function.) See Case 2 in the verification that h^{q^*} is a rank function in the following proof.

Remark 6.16. If we did not include (6a) in the definition of extension, the lemma could fail to hold. Suppose that there is some $i \in F \cap \mathrm{dom}(f^p)$ such that $\sigma := f^q(i) \cap T^p$ properly extends $f^p(i)$. Then $h^q(\sigma) = h^p(\sigma) = \infty$. Consider p^* such that $h^{p^*}(\sigma)$ is sufficiently large (so that p and p^* are appropriate retaggings) but not ∞. Then for any $q^* \leq p^*$, $f^{q^*}(i)$ cannot extend σ (that would imply that $\infty = h^{q^*}(f^{q^*}(i)) \leq h^{q^*}(\sigma) = h^{p^*}(\sigma) < \omega_1^{CK}$). In particular, $f^{q^*}(i)$ cannot be equal to $f^q(i)$, implying that q^* is not an $\omega\gamma$-F-absolute retagging of q.

Remark 6.17. If we do not assume that $F \subseteq \mathrm{dom}(f^p)$, the lemma could fail to hold. Fix $j \in F$ and any $i \neq j$. Suppose that p is such that $f^p(i) = \langle n \rangle$, $j \notin \mathrm{dom}(f^p)$, and $\langle n \rangle$ is unlocked in p. Suppose p^* only differs from p in that $\langle n \rangle$ is locked in p^*. Define $q \leq p$ by adding $f^q(j) = \langle n \rangle$. Then for any $q^* \leq p^*$, (9) ensures that $f^{q^*}(j)$ cannot even extend $\langle n \rangle$, so f^{q^*} and f^q do not agree on F.

Proof of Lemma 6.14. Fix $q \leq p$, p^*, $\gamma < \beta$ as above. Define q^* as follows:

- $T^{q^*} = T^q$;
- $f^{q^*} = f^q$ on F, otherwise $f^{q^*} = f^{p^*}$;
- $l^{q^*} = l^{p^*}$;
- h^{q^*} is defined by cases:

$$h^{q^*}(\tau) = \begin{cases} h^{p^*}(\tau) & \text{if } \tau \in T^p \\ \infty & \text{if } \exists i(\tau \subseteq f^{q^*}(i)) \text{ or } |\tau| \leq 1 \\ h^q(\tau) & \text{if } h^q(\tau) < \omega\gamma \\ \omega\gamma + |\tau|_Q & \text{otherwise} \end{cases}$$

where $Q = \{\sigma \in T^q : h^q(\sigma) \geq \omega\gamma\}$.

We verify that q^* has the desired properties. First we note a fact which will be used twice: if $i \in F \subseteq \operatorname{dom}(f^p)$, then

$$\begin{aligned} f^{q^*}(i) \cap T^p &= f^q(i) \cap T^p && \text{definition of } f^{q^*} \\ &= f^p(i) && i \in \operatorname{dom}(f^p),\ \text{(6a) for } q \leq p \\ &= f^{p^*}(i) && i \in F,\ \text{retagging} \end{aligned}$$

h^{q^*} **is well-defined.** First we show that the second and third case in the definition of h^{q^*} are mutually exclusive. It suffices to show that if $\exists i(\tau \subseteq f^{q^*}(i))$, then $h^q(\tau) \geq \omega\gamma$. There are two cases.

<u>Case 1.</u> If $\tau \subseteq f^q(i)$ for some $i \in F$, then $h^q(\tau) = \infty > \omega\gamma$.

<u>Case 2.</u> Otherwise, $\tau \subseteq f^{p^*}(i)$ for some i. Then $\tau \in T^p$ and $h^{p^*}(\tau) = \infty$, so $h^p(\tau) \geq \omega\beta$ by retagging. It follows that $h^q(\tau) = h^p(\tau) > \omega\gamma$.

Next, we show that the first and second case in the definition of h^{q^*} do not conflict. It suffices to show that if $\tau \in T^p$ and $\exists i(\tau \subseteq f^{q^*}(i))$, then $h^{p^*}(\tau) = \infty$. There are two cases.

<u>Case 1.</u> If $\tau \subseteq f^{p^*}(i)$ for some i, then $h^{p^*}(\tau) = \infty$ as desired.

<u>Case 2.</u> Otherwise, $\tau \subseteq f^{q^*}(i)$ for some $i \in F \subseteq \mathrm{dom}(f^p)$. We noted above that $f^{q^*}(i) \cap T^p = f^{p^*}(i)$. Since $\tau \in T^p$, this case is actually subsumed by Case 1.

Finally, we show that the first and third case in the definition of h^{q^*} do not conflict, i.e., if $\tau \in T^p$ and $h^q(\tau) < \omega\gamma$, then $h^{p^*}(\tau) = h^q(\tau)$. We have that $h^p(\tau) = h^q(\tau) < \omega\gamma$, so by retagging, $h^{p^*}(\tau) = h^p(\tau) = h^q(\tau)$ as desired. We have shown that h^{q^*} is well-defined.

h^{q^*} **is a rank function.** We want to show that for all $\tau \in T^q$ such that $|\tau| \geq 1$, $h^{q^*}(\tau^-) > h^{q^*}(\tau)$. There are $\binom{4}{2} + \binom{4}{1}$ many cases to consider, but we can narrow it down using the following observations:

– h^{q^*} is a rank function on each of the four sets it is piecewise-defined on, so we only need to consider the $\binom{4}{2}$ interactions between the four sets.

– The first and second set (namely T^p and $\{\tau : \exists i(\tau \subseteq f^{q^*}(i))$ or $|\tau| \leq 1\}$ are both downward closed. The third set is upward closed.

– Trivially, $\infty > \omega\gamma + |\gamma|_Q \geq \omega\gamma$.

It follows that we only have two nontrivial cases to consider.

<u>Case 1.</u> $|\tau| > 1$, $\tau \notin T^p$, $\tau^- \in T^p$, $h^q(\tau) < \omega\gamma$. We want to show that $h^{q^*}(\tau) < h^{q^*}(\tau^-)$, i.e., $h^q(\tau) < h^{p^*}(\tau^-)$. If $h^{p^*}(\tau^-) \geq \omega\gamma$, then $h^q(\tau) < \omega\gamma \leq h^{p^*}(\tau^-)$ as desired. Otherwise, $h^{p^*}(\tau^-) < \omega\gamma < \omega\beta$, so $h^p(\tau^-) = h^{p^*}(\tau^-)$ by retagging. But

then $h^q(\tau) < h^q(\tau^-) = h^p(\tau^-) = h^{p^*}(\tau^-)$ as desired.

Case 2. $|\tau| > 1$, $\tau \notin T^p$, $\tau^- \in T^p$, $h^q(\tau) \geq \omega\gamma$, $h^{q^*}(\tau) = \omega\gamma + |\tau|_Q$. If $h^{q^*}(\tau^-) \geq \omega\beta$, then $h^{q^*}(\tau^-) \geq \omega\beta > \omega\gamma + |\tau|_Q = h^{q^*}(\tau)$ as desired. Otherwise, $h^{q^*}(\tau^-) < \omega\beta$. Then

$$\begin{aligned}
h^{q^*}(\tau^-) &= h^{p^*}(\tau^-) && \tau^- \in T^p \\
&= h^p(\tau^-) && \text{retagging} \\
&= h^q(\tau^-) && h^q \supseteq h^p \\
&> h^q(\tau) && h^q \text{ is a rank function} \\
&\geq \omega\gamma + |\tau|_Q && \text{Proposition 6.12} \\
&= h^{q^*}(\tau)
\end{aligned}$$

as desired. This shows that h^{q^*} is a rank function.

q^* **is a condition extending** p^***.** To check that q^* is a condition, it remains to observe that $l^{q^*} = l^q \subseteq \{n : \langle n \rangle \in T^q\} = \{n : \langle n \rangle \in T^{q^*}\}$.

Next, we show that q^* and p^* satisfy (6a). For $i \in \mathrm{dom}(f^{p^*})\backslash F$, $f^{q^*}(i) = f^{p^*}(i)$ so certainly $f^{q^*}(i) \cap T^p = f^{p^*}(i)$ as desired. As for $i \in F$, we noted at the beginning of the proof that $f^{q^*}(i) \cap T^p = f^{p^*}(i)$ as desired.

Finally, q^* and p^* vacuously satisfy (6b) and (9) because

$$\begin{aligned}
\mathrm{dom}(f^{q^*}) &= (F \cap \mathrm{dom}(f^q)) \cup (\mathrm{dom}(f^{p^*})\backslash F) \\
&= F \cup (\mathrm{dom}(f^{p^*})\backslash F) \\
&= \mathrm{dom}(f^{p^*}).
\end{aligned}$$

q **and** q^* **are** $\omega\gamma$**-**F**-absolute retaggings.** It suffices to check that if $h^q(\tau) \geq \omega\gamma$, then $h^{q^*}(\tau) \geq \omega\gamma$. If $\tau \notin T^p$, then $h^{q^*}(\tau) \geq \omega\gamma$ by definition. If $\tau \in T^p$, then

$h^p(\tau) = h^q(\tau) \geq \omega\gamma$. Since p and p^* are $\omega\beta$-F-absolute retaggings, $h^{p^*}(\tau) \geq \omega\gamma$ as well. Hence $h^{q^*}(\tau) = h^{p^*}(\tau) \geq \omega\gamma$ as desired. $\square$

Lemma 6.18. *Let ψ be a ranked formula in $\mathcal{L}_F$. Suppose that p and p^* are $\omega \cdot \mathrm{rk}(\psi)$-$F$-absolute retaggings such that $F \subseteq \mathrm{dom}(f^p)$. Then $p \Vdash \psi$ if and only if $p^* \Vdash \psi$.*

Proof. We proceed by induction on the rank of ψ. The only nontrivial case is when ψ is $\neg\varphi$. Assuming that p^* and p are $\omega \cdot \mathrm{rk}(\neg\varphi)$-$F$-absolute retaggings and that $p^* \Vdash \neg\varphi$, we want to show that $p \Vdash \neg\varphi$, i.e., for all $q \leq p$, $q \nVdash \varphi$. By Lemma 6.14, there is $q^* \leq p^*$ such that q^* and q are $\omega \cdot \mathrm{rk}(\varphi)$-$F$-absolute retaggings. Since $p^* \Vdash \neg\varphi$, we have that $q^* \nVdash \varphi$. Applying the induction hypothesis to q and q^* shows that $q \nVdash \varphi$ as desired. $\square$

Corollary 6.19. *Suppose $p \in \mathbb{P}_{\omega\beta}$ and $\psi \in \mathcal{L}_F$ has rank β. If there is $q \leq p$ such that $q \Vdash \psi$, then there is $q' \leq p$ in $\mathbb{P}_{\omega\beta}$ such that $q' \Vdash \psi$. Therefore, $p \Vdash \neg\psi$ if and only if for all $q \leq p$ in $\mathbb{P}_{\omega\beta}$, $q \nVdash \psi$.*

Proof. First extend q so that $F \subseteq \mathrm{dom}(f^q)$. Then define q' as follows: $T^{q'} = T^q$, $f^{q'} = f^q$, $l^{q'} = l^q$, and define $h^{q'}(\tau) = h^q(\tau)$ if $h^q(\tau) < \omega\beta$ and $h^{q'}(\tau) = \infty$ otherwise. Clearly q' is a condition in $\mathbb{P}_{\omega\beta}$, and q and q' are $\omega\beta$-F-absolute retaggings. Since $F \subseteq \mathrm{dom}(f^q) = \mathrm{dom}(f^{q'})$, by Lemma 6.18, $q' \Vdash \psi$. Finally, in order to check $q' \leq p$, it suffices to check (7). (7) holds because $p \in \mathbb{P}_{\omega\beta}$ and $h^q \supseteq h^p$. $\square$

Corollary 6.20. *If $\psi \in \mathcal{L}_F$ has rank $\beta < \omega_1^{CK}$ and p is a condition with $F \subseteq \mathrm{dom}(f^p)$, then $H_{\beta,\emptyset}$ uniformly computes whether $p \Vdash \psi$.*

Proof. Induction on β using the above corollary. $\square$

Next, we use our basic retagging lemma to analyze which paths on T^G lie in each M_F.

Lemma 6.21. *For each finite $F \subset \omega$, the paths on T^G which lie in M_F are exactly the α_i^G for $i \in F$. Hence $M_\infty \cap [T^G] = \{\alpha_i^G : i \in \omega\}$, but no infinite sequence of distinct α_i^G lies in M_∞.*

Proof. Suppose towards a contradiction that $S = S_{\nu,F,e} \in [T^G]$ is not α_i^G for any $i \in F$. Then there is $\sigma \subset S$ such that $\sigma \not\subset \alpha_i^G$ for any $i \in F$. Without loss of generality, we choose such σ of length ≥ 2. Fix $p \in G$ such that

$$p \Vdash \varphi(\mathbf{S}), \text{ where } \varphi(\mathbf{S}) \text{ is } \mathbf{S} \in [T] \wedge \sigma \subset \mathbf{S} \wedge \forall i \in F(\sigma \not\subset \alpha_i).$$

By genericity, we may assume that $F \subseteq \mathrm{dom}(f^p)$ and $\sigma \in T^p$. Next, fix $\beta < \omega_1^{CK}$ large enough so that $\beta > \omega \cdot \mathrm{rk}(\varphi(\mathbf{S}))$ and $p \in \mathbb{P}_\beta$.

Note that $h^p(\sigma)$ must be ∞. We define p^* which is a β-F-absolute retagging of p, such that $h^{p^*}(\sigma) \in [\beta, \omega_1^{CK})$. Define $T^{p^*} = T^p$, $f^{p^*} = f^p \restriction F$, $l^{p^*} = l^p$, and

$$h^{p^*}(\tau) = \begin{cases} \beta + |\tau|_Q & \tau \supseteq \sigma \wedge h^p(\tau) \geq \beta \\ h^p(\tau) & \text{otherwise} \end{cases},$$

where $Q = \{\tau : \tau \subseteq \sigma \vee (\tau \supseteq \sigma \wedge h^p(\tau) \geq \beta)\}$.

Since $h^p(\sigma) = \infty$ and $|\sigma| \geq 2$, it is easy to see that h^{p^*} is a rank function, and $h^{p^*}(\tau) = \infty$ if $|\tau| < 2$. In order to show that p^* is a condition, it suffices to check that $h^{p^*}(f^{p^*}(i)) = \infty$ for all $i \in \mathrm{dom}(f^{p^*}) = F$. This holds because for all $i \in F$, $\sigma \not\subseteq f^p(i) = f^{p^*}(i)$, so $h^{p^*}(f^{p^*}(i)) = h^p(f^{p^*}(i)) = h^p(f^p(i)) = \infty$.

Finally, it is clear that p^* is a β-F-absolute retagging of p, and hence an $\omega \cdot \mathrm{rk}(\varphi(\mathbf{S}))$-$F$-absolute retagging of p. By Lemma 6.18,

$$p^* \Vdash \mathbf{S} \in [T] \wedge \sigma \subset \mathbf{S},$$

which is impossible because $h^{p^*}(\sigma) < \omega_1^{CK}$. □

Lemma 6.22. *M_∞ does not satisfy finite-Σ^1_1-AC.*

Proof. For each n, let T_n be the subtree of T^G passing through $\langle n\rangle$. By genericity (given p, we can expand l^p to include n), each $\langle n\rangle$ is eventually locked, so there are only finitely many α_i^G passing through $\langle n\rangle$. Also by genericity (if there is no i such that $f^p(i)$ passes through $\langle n\rangle$, we can choose some fresh j and add $f^p(j) = \langle n\rangle$ to p), there must be some α_i^G passing through $\langle n\rangle$.

Now, we showed in Lemma 6.21 that the paths on T^G in M_∞ are exactly $\{\alpha_i^G : i \in \omega\}$. So M_∞ thinks that $\langle T_n\rangle_n$ is an instance of finite-Σ^1_1-AC.

However, we showed in Lemma 6.21 that M_∞ does not contain any infinite sequence of distinct α_i^G. So M_∞ does not contain any finite-Σ^1_1-AC solution to $\langle T_n\rangle_n$. □

6.3.6 Analyzing the forcing relation for Σ-over-$\mathcal{L}_F$ formulas

In order to analyze the forcing relation for Σ-over-$\mathcal{L}_F$ formulas, we define a stronger retagging notion which places restrictions on locks.

Definition 6.23. We define $\mathrm{Ret}_{\leq}(\beta, F, p, p^*)$ if:

- p and p^* are β-F-absolute retaggings;
- $\{n : \exists i(f^p(i)(0) = n)\} \supseteq \{n : \exists i(f^{p^*}(i)(0) = n)\}$;
- $l^p \supseteq l^{p^*}$.

We make some observations:

– $\mathrm{Ret}_{\leq}(\beta, F, \cdot, \cdot)$ is reflexive and transitive, but not symmetric. This asymmetry is not essential; we could have enforced equality in the second and third condition and proved Lemma 6.24.

– The dependence of $\mathrm{Ret}_{\leq}(\beta, F, \cdot, \cdot)$ on f^p and f^{p^*} is not "local" to F: whether $\mathrm{Ret}_{\leq}(\beta, F, p, p^*)$ holds depends on more than just $f^p \restriction F$ and $f^{p^*} \restriction F$.

We prove a basic retagging lemma.

Lemma 6.24. *Suppose that* $\mathrm{Ret}_{\leq}(\omega\beta, F, p, p^*)$. *Then for all* $q \leq p$ *and all* $\gamma < \beta$, *there exists* $q^* \leq p^*$ *such that* $\mathrm{Ret}_{\leq}(\omega\gamma, F, q, q^*)$.

Proof. Fix $q \leq p$, p^*, and $\gamma < \beta$ as above. Define q^* as follows:

– $T^{q^*} = T^q$;

– $f^{q^*} = f^q$ on F and $f^{q^*} = f^{p^*}$ on $\mathrm{dom}(f^{p^*}) \backslash F$;

– $l^{q^*} = l^q$;

– h^{q^*} is defined by cases:

$$h^{q^*}(\tau) = \begin{cases} h^{p^*}(\tau) & \text{if } \tau \in T^p \\ \infty & \text{if } \exists i(\tau \subseteq f^{q^*}(i)) \text{ or } |\tau| \leq 1 \\ h^q(\tau) & \text{if } h^q(\tau) < \omega\gamma \\ \omega\gamma + |\tau|_Q & \text{otherwise} \end{cases}$$

where $Q = \{\sigma \in T^q : h^q(\sigma) \geq \omega\gamma\}$.

We verify that q^* has the desired properties.

h^{q^*} **is well-defined.** The proof is almost the same as that for Lemma 6.14, except where we show that the first and second case in the definition of h^{q^*} do not conflict. Since we do not assume here that $F \subseteq \text{dom}(f^p)$, we need to consider the following situation, which was impossible in Lemma 6.14.

Suppose that $\tau \subseteq f^{q^*}(i)$ for some $i \in (F \cap \text{dom}(f^q))\backslash\text{dom}(f^p)$. By definition $f^{q^*}(i) = f^q(i)$. By (6b) for $q \leq p$, $|f^q(i) \cap T^p| \leq 1$. Since $\tau \in T^p$, it follows that $|\tau| \leq 1$. So $h^{p^*}(\tau) = \infty$ as desired.

q^* **is a condition.** The proof is the same as that for Lemma 6.14.

q^* **extends** p^***.** We start by showing that q^* and p^* satisfy (6a). For $i \in \text{dom}(f^{p^*})\backslash F$, $f^{q^*}(i) = f^{p^*}(i) \in T^p$ so certainly $f^{q^*}(i) \cap T^p = f^{p^*}(i)$ as desired. As for $i \in \text{dom}(f^{p^*}) \cap F = \text{dom}(f^p) \cap F$, we showed earlier (in the proof that h^{q^*} is well-defined) that $f^{q^*}(i) \cap T^p = f^{p^*}(i)$ as desired.

For (6b) and (9), let $i \in \text{dom}(f^{q^*})\backslash\text{dom}(f^{p^*}) = (F \cap \text{dom}(f^q))\backslash\text{dom}(f^p)$. We showed earlier (in the proof that h^{q^*} is well-defined) that $|f^{q^*}(i) \cap T^p| \leq 1$, showing (6b).

As for (9), suppose that $f^{q^*}(i)(0) = n$ and there is j such that $f^{p^*}(j)(0) = n$. We want to show that $n \notin l^{p^*}$. First, by definition of q^*, $f^{q^*}(i) = f^q(i)$. Second, by the second condition in $\text{Ret}_{\leq}(\omega\beta, F, p, p^*)$, there is some j' such that $f^p(j')(0) = n$. Hence we may apply (9) for $q \leq p$ to conclude that $n \notin l^p$. Finally, by the third condition in $\text{Ret}_{\leq}(\omega\beta, F, p, p^*)$, $l^p \supseteq l^{p^*}$. Thus $n \notin l^{p^*}$ as desired.

q **and** q^* **satisfy** $\text{Ret}_{\leq}(\omega\gamma, F, q, q^*)$**.** The proof that $h^q(\tau) \geq \omega\gamma$ implies $h^{q^*}(\tau) \geq \omega\gamma$ is the same as that for Lemma 6.14. Hence q and q^* are $\omega\gamma$-F-

absolute retaggings. Next, observe that

$$\begin{aligned}
&\{n : \exists i(f^{q^*}(i)(0) = n)\} \\
\subseteq\ &\{n : \exists i(f^{q}(i)(0) = n)\} \cup \{n : \exists i(f^{p^*}(i)(0) = n)\} \\
\subseteq\ &\{n : \exists i(f^{q}(i)(0) = n)\} \cup \{n : \exists i(f^{p}(i)(0) = n)\} && \mathrm{Ret}_{\leq}(\omega\beta, F, p, p^*) \\
\subseteq\ &\{n : \exists i(f^{q}(i)(0) = n)\} && \text{(6a) for } q \leq p.
\end{aligned}$$

Finally, $l^{q^*} = l^q$ by definition. □

Next, we study a retagging notion even stronger than that in Definition 6.23 and show that it respects the forcing relation for Σ-over-$\mathcal{L}_F$ formulas (Lemma 6.26).

Definition 6.25. We abbreviate $\mathrm{Ret}_{\leq}(\beta, \mathrm{dom}(f^p), p, p^*)$ by $\mathrm{Ret}_{\leq}(\beta, p, p^*)$. Equivalently, $\mathrm{Ret}_{\leq}(\beta, p, p^*)$ if:

- p and p^* are β-$\mathrm{dom}(f^p)$-absolute retaggings;
- $\{n : \exists i(f^p(i)(0) = n)\} = \{n : \exists i(f^{p^*}(i)(0) = n)\}$;
- $l^p \supseteq l^{p^*}$.

We make some observations:

- $\mathrm{Ret}_{\leq}(\beta, F, \cdot, \cdot)$ is reflexive and transitive, but not symmetric.
- In the second condition, we have equality (instead of $\supseteq$) because the first condition of $\mathrm{Ret}_{\leq}(\beta, p, p^*)$ implies that $f^p \subseteq f^{p^*}$.

Our goal is then to prove:

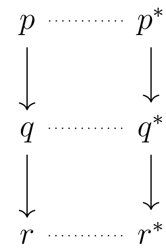

Figure 6.1: Arrows correspond to extension in the forcing. Dotted lines correspond to some notion of retagging, which will be made precise in the proof of Lemma 6.26.

Lemma 6.26. *Suppose that ψ is Σ-over-$\mathcal{L}_F$, $p \Vdash \psi^\mu$, and $F \subseteq \operatorname{dom}(f^p)$. If $\operatorname{Ret}_{\leq}(\omega \cdot \operatorname{rk}(\psi^\mu) + \omega^2, p, p^*)$, then $p^* \Vdash \psi^\mu$ as well.*

The proof has two main components: a retagging lemma (Lemma 6.27), and a class of automorphisms of our forcing $\mathbb{P}$, obtained by using permutations of ω to permute the domain of f^p. Before presenting the proof, we discuss our strategy, which is illustrated in Figure 6.1. For simplicity, suppose for now that ψ is $\exists X \varphi(X)$, where φ is ranked and F-restricted. Suppose that $p \Vdash \psi^\mu$, $F \subseteq \operatorname{dom}(f^p)$, and $\operatorname{Ret}_{\leq}(\omega \cdot \operatorname{rk}(\psi^\mu) + \omega^2, p, p^*)$. Given $q^* \leq p^*$, we want to construct $r^* \leq q^*$ and $\mathbf{S} \in C^\mu$ such that $r^* \Vdash \varphi^\mu(\mathbf{S})$.

The first step in our plan is to construct $q \leq p$ which is a retagging of q^* (for some appropriate notion of retagging). Next, since $p \Vdash \psi^\mu$, there must be some $r \leq q$ and some $\mathbf{S} \in C^\mu$ such that $r \Vdash \varphi^\mu(\mathbf{S})$. By extending r, we may assume without loss of generality that $\operatorname{dom}(\mathbf{S}) \subseteq \operatorname{dom}(f^r)$. Finally, if we could construct $r^* \leq q^*$ which is an $\omega \cdot \operatorname{rk}(\varphi^\mu)$-$(F \cup \operatorname{dom}(\mathbf{S}))$-absolute retagging of r, then we could conclude by Lemma 6.18 that $r^* \Vdash \varphi^\mu(\mathbf{S})$ as desired.

What properties does q have to satisfy in order for us to construct r^* as above? For one, there cannot be any i such that $f^q(i)$ and $f^{q^*}(i)$ are defined but different. For if $i \in \operatorname{dom}(\mathbf{S})$, there is no r^* extending q^* which satisfies $f^{r^*} \restriction (F \cup \operatorname{dom}(\mathbf{S})) =$

$f^r \restriction (F \cup \mathrm{dom}(\mathbf{S}))$! (This follows from (6a) for $r^* \leq q^*$ and $r \leq q$.) Therefore we will construct q such that f^q and f^{q^*} agree on $\mathrm{dom}(f^q) \cap \mathrm{dom}(f^{q^*})$.

As for $i \in \mathrm{dom}(f^{q^*}) \backslash \mathrm{dom}(f^q)$, we can avoid problems using automorphisms of our forcing. As we argue later, we may permute r and $\mathbf{S}$ to ensure that $\mathrm{dom}(f^r) \cap \mathrm{dom}(f^{q^*}) = \mathrm{dom}(f^q)$ (while preserving the facts that $r \leq q$ and $r \Vdash \varphi^\mu(\mathbf{S})$). Then $\mathrm{dom}(f^{q^*}) \backslash \mathrm{dom}(f^q)$ and $\mathrm{dom}(\mathbf{S})$ are disjoint, so we are not obliged to (and indeed will not) define $f^{r^*}(i) = f^r(i)$.

How about $i \in \mathrm{dom}(f^q) \backslash \mathrm{dom}(f^{q^*})$? Consider the situation where $r = q$ and $i \in \mathrm{dom}(\mathbf{S})$. Then if $f^{r^*} \restriction (F \cup \mathrm{dom}(\mathbf{S})) = f^r \restriction (F \cup \mathrm{dom}(\mathbf{S}))$, we must have $f^{r^*}(i) = f^q(i)$. Hence we must be able to extend q^* by defining $f^{r^*}(i) = f^q(i)$. But since $i \in \mathrm{dom}(f^{r^*}) \backslash \mathrm{dom}(f^{q^*})$, $f^{r^*}(i)$ is constrained by (6b) and (9) for $r^* \leq q^*$. This creates multiple problems, so we will avoid this by constructing q such that this cannot happen, i.e., $\mathrm{dom}(f^q) \subseteq \mathrm{dom}(f^{q^*})$. Together with our earlier commitment that f^q and f^{q^*} agree on $\mathrm{dom}(f^q) \cap \mathrm{dom}(f^{q^*})$, this means that we want to construct q such that $f^q \subseteq f^{q^*}$.

Unfortunately, we cannot always avoid (6b) and (9). Consider a situation where r extends q by adding some $f^r(i)$, $i \in \mathrm{dom}(f^r) \backslash \mathrm{dom}(f^q)$. Suppose that $i \in \mathrm{dom}(\mathbf{S})$. Then we must define $f^{r^*}(i) = f^r(i)$. Since $\mathrm{dom}(f^r) \cap \mathrm{dom}(f^{q^*}) = \mathrm{dom}(f^q)$, we have that $i \in \mathrm{dom}(f^{r^*}) \backslash \mathrm{dom}(f^{q^*})$. Hence $f^{r^*}(i)$ is constrained by (6b) and (9) for $r^* \leq q^*$.

However, observe that $f^r(i)$ is constrained by (6b) and (9) for $r \leq q$! (6b) for $r \leq q$ demands that $|f^r(i) \cap T^q| \leq 1$. Since $f^{r^*}(i) = f^r(i)$ and $T^q = T^{q^*}$, (6b) for $r^* \leq q^*$ holds as well. Similarly, we will argue that (9) for $r^* \leq q^*$ follows from (9) for $r \leq q$. What is needed for that argument is exactly the following statement about q and q^*:

$P(q, q^*)$: If $n \in l^{q^*}$ and there is some $i \in \mathrm{dom}(f^{q^*})$ such that $f^{q^*}(i)(0) = n$, then $n \in l^q$ and there is some $j \in \mathrm{dom}(f^q)$ such that $f^q(j)(0) = n$.

Observe that if $l^q \supseteq l^{q^*}$ and $\{n : \exists i(f^q(i)(0) = n)\} \supseteq \{n : \exists i(f^{q^*}(i)(0) = n)\}$, then $P(q, q^*)$ certainly holds. This partially justifies the second and third condition in Definition 6.25. To fully justify those conditions, we need to bring p and p^* back into the picture. Stronger assumptions, such as those in Definition 6.25, are needed on p and p^* in order to ensure that we can always construct q satisfying the desired properties.

There is another restriction that we need to impose on p and p^*. Suppose that there is some $i \in \mathrm{dom}(f^p) \setminus \mathrm{dom}(f^{p^*})$. Define $q^* \leq p^*$ by adding $f^p(i)(0)$ to l^{q^*}. Next, suppose that we manage to construct some $q \leq p$. We then obtain some $r \leq q$ and some $\mathbf{S}$. Suppose that $i \in \mathrm{dom}(\mathbf{S})$. Then we are obliged to define $f^{r^*}(i) = f^r(i)$. But $i \notin \mathrm{dom}(f^{q^*})$ and $f^{r^*}(i)(0) = f^p(i)(0) \in l^{q^*}$, violating (9). So there is no $r^* \leq q^*$ satisfying the desired properties. Therefore we must require that $\mathrm{dom}(f^p) \subseteq \mathrm{dom}(f^{p^*})$, justifying the first condition in Definition 6.25.

This concludes our preliminary discussion of the proof of Lemma 6.26. We proceed to present the details. First, we prove a retagging lemma. It looks similar to Lemma 6.24, but yields a stronger result in the case where $F = \mathrm{dom}(f^{p^*})$ (in this case Lemma 6.24 only yields some $q^* \leq p^*$ such that $\mathrm{Ret}_{\leq}(\omega\gamma, \mathrm{dom}(f^{p^*}), q^*, q)$.)

Lemma 6.27. *Suppose that* $\mathrm{Ret}_{\leq}(\omega\beta, p^*, p)$*,* $q \leq p$*, and* $\gamma < \beta$*. Then there is* $q^* \leq p^*$ *such that* $\mathrm{Ret}_{\leq}(\omega\gamma, q^*, q)$*.*

Proof. Fix $q \leq p$, p^*, $\gamma < \beta$ as above. Define q^* as follows:

– $T^{q^*} = T^q$;

– $f^{q^*} = f^q \upharpoonright (H \cup \text{dom}(f^{p^*}))$, where

$$H = \text{dom}(f^q) \backslash \text{dom}(f^p);$$

– $l^{q^*} = l^q$;

– h^{q^*} is defined by cases:

$$h^{q^*}(\tau) = \begin{cases} h^{p^*}(\tau) & \text{if } \tau \in T^p \\ \infty & \text{if } \exists i(\tau \subseteq f^{q^*}(i)) \text{ or } |\tau| \leq 1 \\ h^q(\tau) & \text{if } h^q(\tau) < \omega\gamma \\ \omega\gamma + |\tau|_Q & \text{otherwise} \end{cases}$$

where $Q = \{\sigma \in T^q : h^q(\sigma) \geq \omega\gamma\}$.

h^{q^*} **is well-defined.** First observe that since $f^{q^*} \subseteq f^q$, the second and third case are mutually exclusive.

Next, assuming that $\tau \in T^p$ and $\exists i(\tau \subseteq f^{q^*}(i))$, we show that $h^{p^*}(\tau) = \infty$.

If $i \in \text{dom}(f^{p^*}) \subseteq \text{dom}(f^p)$, then $\tau \subseteq f^{q^*}(i) \cap T^p = f^q(i) \cap T^p = f^p(i)$ by (6a) for $q \leq p$. Next, $f^p(i) = f^{p^*}(i)$ by retagging, so $h^{p^*}(\tau) = \infty$ as desired.

Otherwise, $i \in H = \text{dom}(f^q) \backslash \text{dom}(f^p)$. Then $|f^{q^*}(i) \cap T^p| = |f^q(i) \cap T^p| \leq 1$ by (6b) for $q \leq p$, so $h^{p^*}(\tau) = \infty$ as desired.

Next, assuming that $\tau \in T^p$ and $h^q(\tau) < \omega\gamma$, we show that $h^{p^*}(\tau) = h^q(\tau)$: $h^p(\tau) = h^q(\tau) < \omega\gamma$, so by retagging, $h^{p^*}(\tau) = h^p(\tau) = h^q(\tau)$. We have shown that h^{q^*} is well-defined.

q^* **is a condition.** The proof is the same as that for Lemma 6.14.

q^* **extends** p^***.** (5) holds because of (5) for $q \leq p$. For (6a): if $i \in \mathrm{dom}(f^{p^*}) \subseteq \mathrm{dom}(f^p)$, we have that

$$\begin{aligned} f^{q^*}(i) \cap T^p &= f^q(i) \cap T^p && i \in \mathrm{dom}(f^{p^*}) \\ &= f^p(i) && i \in \mathrm{dom}(f^p),\ \text{(6a) for } q \leq p \\ &= f^{p^*}(i) && f^{p^*} \subseteq f^p \end{aligned}$$

as desired. For (6b), observe that $\mathrm{dom}(f^{q^*})\backslash\mathrm{dom}(f^{p^*}) = \mathrm{dom}(f^q)\backslash\mathrm{dom}(f^p)$, so (6b) follows from (6b) for $q \leq p$. (7) holds by definition. (8) follows from (8) for $q \leq p$ and $\mathrm{Ret}_{\leq}(\omega\beta, p^*, p)$: $l^{q^*} = l^q \supseteq l^p = l^{p^*}$.

Finally, (9) follows from (9) for $q \leq p$ and $\mathrm{Ret}_{\leq}(\omega\beta, p^*, p)$: suppose $n \in l^{p^*}$ and there is j such that $f^{p^*}(j)(0) = n$. We have that $l^{p^*} = l^p$ and $f^{p^*} \subseteq f^p$, so $n \in l^p$ and $f^p(j)(0) = n$. By (9) for $q \leq p$, there is no $i \in \mathrm{dom}(f^q)\backslash\mathrm{dom}(f^p)$ such that $f^q(i)(0) = n$. Since $f^{q^*} \subseteq f^q$ and $\mathrm{dom}(f^{q^*})\backslash\mathrm{dom}(f^{p^*}) = \mathrm{dom}(f^q)\backslash\mathrm{dom}(f^p)$, there is no $i \in \mathrm{dom}(f^{q^*})\backslash\mathrm{dom}(f^{p^*})$ such that $f^{q^*}(i)(0) = n$, as desired.

$\{n : \exists i[f^q(i)(0) = n]\}$ **equals** $\{n : \exists i[f^{q^*}(i)(0) = n]\}$**.** The backward inclusion holds because $f^{q^*} \subseteq f^q$. For the forward inclusion:

$$\begin{aligned} & \{n : \exists i \in \mathrm{dom}(f^p)\backslash\mathrm{dom}(f^{p^*})[f^q(i)(0) = n]\} \\ &= \{n : \exists i \in \mathrm{dom}(f^p)\backslash\mathrm{dom}(f^{p^*})[f^p(i)(0) = n]\} && \text{(6a) for } q \leq p \\ &\subseteq \{n : \exists i \in \mathrm{dom}(f^{p^*})[f^{p^*}(i)(0) = n]\} && \mathrm{Ret}_{\leq}(\omega\beta, p^*, p) \\ &= \{n : \exists i \in \mathrm{dom}(f^{p^*})[f^p(i)(0) = n]\} && f^{p^*} \subseteq f^p \\ &= \{n : \exists i \in \mathrm{dom}(f^{p^*})[f^q(i)(0) = n]\} && \text{(6a) for } q \leq p \\ &\subseteq \{n : \exists i[f^{q^*}(i)(0) = n]\} && f^q \restriction \mathrm{dom}(f^{p^*}) \subseteq f^{q^*}. \end{aligned}$$

q **and** q^* **satisfy** $\mathrm{Ret}_{\leq}(\omega\gamma, q^*, q)$**.** We omit the routine proof that q^* and q are

$\omega\gamma$-dom(f^{q^*})-absolute retaggings. We showed above that $\{n : \exists i[f^q(i)(0) = n]\}$ equals $\{n : \exists i[f^{q^*}(i)(0) = n]\}$. Finally, $l^{q^*} = l^q$ by definition. $\square$

Next, observe that every permutation π of ω induces an automorphism $\hat{\pi}$ of our forcing, defined by $T^{\hat{\pi}(p)} = T^p$, $f^{\hat{\pi}(p)}(\pi(i)) = f^p(i)$, $h^{\hat{\pi}(p)} = h^p$, and $l^{\hat{\pi}(p)} = l^p$.

π also induces a bijection on formulas in $\mathcal{L}_\infty$: for any ψ in $\mathcal{L}_\infty$, $\pi\psi$ is defined to be the formula obtained from ψ by replacing α_i with $\alpha_{\pi(i)}$, $H_{\nu,F}$ with $H_{\nu,\pi''F}$, $S_{\nu,F,e}$ with $S_{\nu,\pi''F,e}$, X^ν_F with $X^\nu_{\pi''F}$, and X_F with $X_{\pi''F}$.

Lemma 6.28. *Suppose that $p \Vdash \psi$. If π is a permutation of ω, then $\hat{\pi}(p) \Vdash \pi\psi$.*

Proof. Induction on rank of ψ. $\square$

A typical application of Lemma 6.28 is as follows.

Corollary 6.29. *Let $\varphi \in \mathcal{L}_F$. Suppose that p and $H \subset_f \omega$ are such that $F \subseteq \mathrm{dom}(f^p) \subseteq H$. If there is $q \leq p$ such that $q \Vdash \varphi$, then there is $q' \leq p$ such that $q' \Vdash \varphi$ and $\mathrm{dom}(f^{q'}) \cap H = \mathrm{dom}(f^p)$.*

Proof. Let π permute q to q' by fixing $\mathrm{dom}(f^p)$ and moving $\mathrm{dom}(f^q)\backslash\mathrm{dom}(f^p)$ to some set disjoint from H. Lemma 6.28 implies that $q' \Vdash \pi\varphi$. Since π fixes $F \subseteq \mathrm{dom}(f^p)$ and $\varphi \in \mathcal{L}_F$, $\pi\varphi$ is the same as φ. Finally, it is easy to see that $q' \leq p$. $\square$

We are ready to show that our retagging notion respects the forcing relation for Σ-over-$\mathcal{L}_F$ formulas.

Proof of Lemma 6.26. We prove the following statement by induction on k:

> Suppose that ψ is Σ-over-$\mathcal{L}_F$ and can be constructed from ranked F-restricted formulas in k steps. Suppose that $p \Vdash \psi^\mu$ and $F \subseteq \mathrm{dom}(f^p)$. If $\mathrm{Ret}_\leq(\omega \cdot \mathrm{rk}(\psi^\mu) + \omega \cdot 2k, p, p^*)$, then $p^* \Vdash \psi^\mu$ as well.

The case $k = 0$ follows from Lemma 6.18. Assume that the above statement holds for k. Fix ψ which is Σ-over-$\mathcal{L}_F$ and can be constructed from ranked F-restricted formulas in $k + 1$ steps.

The only nontrivial case is where ψ is of the form $\exists X \varphi(X)$, and φ can be constructed from ranked F-restricted formulas in k steps.

Suppose that $p \Vdash \psi^\mu$. Let σ denote $\mathrm{rk}(\psi^\mu)$. Suppose that $\mathrm{Ret}_\leq(\omega\sigma + \omega \cdot (2k + 2), p, p^*)$. Given $q^* \leq p^*$, we have to construct $r^* \leq q^*$ and $S \in C^\mu$ such that $r^* \Vdash \varphi^\mu(S)$. This plan is illustrated in Figure 6.1.

First, by Lemma 6.27, there is $q \leq p$ such that $\mathrm{Ret}_\leq(\omega\sigma + \omega \cdot (2k + 1), q, q^*)$. Since $p \Vdash \psi^\mu$, there are $r \leq q$ and $S \in C^\mu$ such that $r \Vdash \varphi^\mu(S)$.

Without loss of generality, we may assume that $\mathrm{dom}(S) \subseteq \mathrm{dom}(f^r)$ and $\mathrm{dom}(f^{q^*}) \cap \mathrm{dom}(f^r) = \mathrm{dom}(f^q)$. This is arranged by first extending r to ensure that $\mathrm{dom}(S) \subseteq \mathrm{dom}(f^r)$. Then, consider a permutation π which fixes $\mathrm{dom}(f^q)$ (and hence F) and moves $\mathrm{dom}(f^r) \setminus \mathrm{dom}(f^q)$ away from $\mathrm{dom}(f^{q^*})$. We show that πr and πS have the desired properties. Firstly, it is easy to see that $\pi r \leq q$. Secondly, Lemma 6.28 implies that $\pi r \Vdash \pi\varphi^\mu(\pi S)$. Since $F \subseteq \mathrm{dom}(f^q)$ and π fixes $\mathrm{dom}(f^q)$, $\pi\varphi$ is the same as φ. Hence $\pi r \Vdash \varphi^\mu(\pi S)$. Thirdly, $\mathrm{dom}(\pi S) = \pi(\mathrm{dom}(S)) \subseteq \pi(\mathrm{dom}(f^r)) = \mathrm{dom}(f^{\pi r})$, so πr and πS satisfy the desired properties.

To complete the proof, observe that because $\mathrm{Ret}_\leq(\omega\sigma + \omega \cdot (2k+1), \mathrm{dom}(f^q), q, q^*)$ and $\mathrm{dom}(f^{q^*}) \cap \mathrm{dom}(f^r) = \mathrm{dom}(f^q)$, we in fact have that $\mathrm{Ret}_\leq(\omega\sigma + \omega \cdot (2k +$

$1), \mathrm{dom}(f^r), q, q^*)$. Now we apply Lemma 6.24 to obtain $r^* \leq q^*$ such that $\mathrm{Ret}_{\leq}(\omega\sigma + \omega \cdot 2k, \mathrm{dom}(f^r), r, r^*)$. Note that $\varphi(S)$ is Σ-over-$\mathcal{L}_{F \cup \mathrm{dom}(S)}$ and can be constructed from ranked $(F \cup \mathrm{dom}(S))$-restricted formulas in k steps, $r \Vdash \varphi^\mu(S)$, and $F \cup \mathrm{dom}(S) \subseteq \mathrm{dom}(f^r)$. The inductive hypothesis then implies that $r^* \Vdash \varphi^\mu(S)$, as desired. □

Now we may prove a boundedness result. Its statement and proof is almost the same as that of Montalbán [31, Lemma 2.9]. One important difference is that we cannot use Lemma 6.14 in our proof (because it only applies when $F \subseteq \mathrm{dom}(f^p)$), while Montalbán can use his Lemma 2.5. Instead, we use Lemma 6.24.

Lemma 6.30. *Suppose that ψ is Σ-over-$\mathcal{L}_F$. Then, for conditions p such that $F \subseteq \mathrm{dom}(f^p)$:*

- *if $p \Vdash \psi$, there is $\mu < \omega_1^{CK}$ such that for all $\rho \in [\mu, \omega_1^{CK})$, $p \Vdash \psi^\rho$;*
- *given p, ψ and $\rho < \omega_1^{CK}$, we can compute whether $p \Vdash \psi^\rho$ uniformly in some $H_{\mu,F}$ (μ depending on ψ^ρ).*

Proof. We proceed by induction on the number of steps it takes to construct ψ from ranked F-restricted formulas. The base case holds by Corollary 6.20.

The only nontrivial case is when ψ has the form $\exists X \varphi(X)$. Since $p \Vdash \psi$, for every $q \leq p$, there is $r \leq q$ and $\mathbf{S}$ such that $r \Vdash \varphi(\mathbf{S})$. By extending r, we may ensure that $F \cup \mathrm{dom}(\mathbf{S}) \subseteq \mathrm{dom}(f^r)$. Then, by the induction hypothesis applied to $r \Vdash \varphi(\mathbf{S})$, $r \Vdash \varphi^\rho(\mathbf{S})$ for all sufficiently large $\rho < \omega_1^{CK}$.

Hence, for each $q \leq p$, there is $\gamma_q < \omega_1^{CK}$, $r \leq q$ in $\mathbb{P}_{\gamma_q}$, and $\mathbf{S} \in C^{\gamma_q}$ such that $r \Vdash \varphi^{\gamma_q}(\mathbf{S})$. By the induction hypothesis, we can hyperarithmetically search for the least such $\gamma_q < \omega_1^{CK}$.

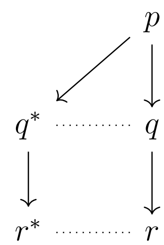

Figure 6.2: Arrows correspond to extension in the forcing. Dotted lines correspond to retaggings.

By boundedness, for each $\beta < \omega_1^{CK}$, there must be $\gamma < \omega_1^{CK}$ such that for every $q \leq p$ in $\mathbb{P}_\beta$, there is $\gamma_q < \gamma$, r, and $\mathbf{S}$ as above. For later purposes, we will choose γ sufficiently large such that $\omega \cdot \mathrm{rk}(\varphi^{\gamma_q}(\mathbf{S})) + \omega^2 + \omega \leq \gamma$.

In fact, we can search for such γ hyperarithmetically, so by boundedness, we can construct some limit $\mu < \omega_1^{CK}$ by recursion along ω, such that (1) $p \in \mathbb{P}_\mu$; (2) for every $\beta < \mu$, we can find such $\gamma < \mu$.

We show that $p \Vdash \psi^\mu$: given $q \leq p$, we want to construct $r \leq q$ and $\mathbf{S} \in C^\mu$ such that $r \Vdash \varphi^\mu(\mathbf{S})$. Our plan for doing so is illustrated in Figure 6.2.

First we define q^* such that $\mathrm{Ret}_\leq(\mu, H, q^*, q)$ for any H: $T^{q^*} = T^q$, $f^{q^*} = f^q$, $l^{q^*} = l^q$, and for all σ such that $h^q(\sigma) \geq \mu$, we define $h^{q^*}(\sigma) = \infty$, otherwise define $h^{q^*}(\sigma) = h^q(\sigma)$. Clearly q^* is a condition in $\mathbb{P}_\mu$, q^* extends p, and $\mathrm{Ret}_\leq(\mu, H, q^*, q)$ for any H.

Since μ is a limit and $\mathrm{dom}(h^{q^*}) = T^{q^*}$ is finite, $q^* \in \mathbb{P}_\beta$ for some $\beta < \mu$. By construction of μ, there exist $\gamma_{q^*} < \gamma < \mu$, $r^* \leq q^*$ in $\mathbb{P}_{\gamma_{q^*}}$, and $\mathbf{S} \in C^{\gamma_{q^*}}$ such that $\omega \cdot \mathrm{rk}(\varphi^{\gamma_q}(\mathbf{S})) + \omega^2 + \omega \leq \gamma$ and $r^* \Vdash \varphi^{\gamma_{q^*}}(\mathbf{S})$. We may extend r^* to ensure that $F \cup \mathrm{dom}(\mathbf{S}) \subseteq \mathrm{dom}(f^{r^*})$.

Next, since $\omega \cdot \mathrm{rk}(\varphi^{\gamma_q}(\mathbf{S})) + \omega^2 + \omega \leq \gamma < \mu$ and $\mathrm{Ret}_\leq(\mu, \mathrm{dom}(f^{r^*}), q^*, q)$, by Lemma 6.24, there is some $r \leq q$ such that $\mathrm{Ret}_\leq(\omega \cdot \mathrm{rk}(\varphi^{\gamma_{q^*}}(\mathbf{S})) + \omega^2, \mathrm{dom}(f^{r^*}), r^*, r)$.

Since $F \cup \mathrm{dom}(\mathbf{S}) \subseteq \mathrm{dom}(f^{r^*})$, by Lemma 6.26, $r \Vdash \varphi^{\gamma_{q^*}}(\mathbf{S})$, and so $r \Vdash \varphi^{\mu}(\mathbf{S})$ as desired.

The above argument also shows us how to compute whether $p \Vdash \exists X^\rho \varphi^\rho(X^\rho)$, in a hyperarithmetic way. Let $\mu < \omega_1^{CK}$ be larger than $\omega \cdot \mathrm{rk}(\varphi^\rho(\mathbf{S})) + \omega^2 + \omega$ for any $\mathbf{S} \in C^\rho$. Then $p \Vdash \exists X^\rho \varphi^\rho(X^\rho)$ if and only if for all $q \leq p$ in $\mathbb{P}_\mu$, there is some $r \leq q$ in $\mathbb{P}_\mu$ and $\mathbf{S} \in C^\rho$ such that $r \Vdash \varphi^\rho(\mathbf{S})$. □

Just as in [31, Lemma 2.9], we conclude that

Corollary 6.31. *For each F, $M_F \models \Sigma_1^1$-AC_0. It follows that $M_F = \mathrm{HYP}(T^G \oplus \langle \alpha_i^G \rangle_{i \in F})$.*

Proof. Suppose that $M_F \models \forall n \exists X \varphi(n, X)$, where $\varphi(n, X)$ is arithmetic. Then there is some $p \in G$ such that $p \Vdash \forall n \exists X \varphi(n, X)$. By Lemma 6.30, fix some $\mu < \omega_1^{CK}$ such that $p \Vdash \forall n \exists X^\mu \varphi(n, X^\mu)$. Next, by Corollary 6.20, for each n, we can use $H_{\mu+\omega,F}$ to search for the least e_n such that $p \Vdash \varphi(\mathbf{n}, \mathbf{S}_{\mu,F,e_n})$. Hence $(S_{\mu,F,e_n})_n$ lies in M_F, as desired. □

6.3.7 M_∞ satisfies Δ_1^1-comprehension

Definition 6.32. We say that $h : T \to \omega_1^{CK} \cup \{\infty\}$ is *ν-good* if $T \subset T^G$ and for all $\sigma \in T$:

$$h^G(\sigma) < \nu \quad \Rightarrow \quad h(\sigma) = h^G(\sigma)$$
$$h^G(\sigma) \geq \nu \quad \Leftrightarrow \quad h(\sigma) \geq \nu.$$

To complete the proof of Theorem 6.11, we show that:

Lemma 6.33. $M_\infty \models \Delta^1_1\text{-}\mathsf{CA}_0$.

Proof. Suppose that $\varphi(n)$ and $\psi(n)$ are Σ-over-$\mathcal{L}_F$ with only n free and $M_\infty \models \forall n(\psi(n) \leftrightarrow \neg\varphi(n))$. We want to define $D \in M_\infty$ such that

$$M_\infty \models \forall n(\psi(n) \leftrightarrow n \in D).$$

A naive attempt is to consider

$$\{n : \exists q \in G(q \Vdash \psi(\mathbf{n}))\},$$

but there are two obstacles preventing us from showing that the above set lies in M_∞:

- M_∞ does not contain G so we cannot search over $q \in G$;
- we do not know that deciding whether $q \Vdash \psi(\mathbf{n})$ is hyperarithmetic.

To overcome the first obstacle, we use retagging to change the scope of our search to a class of conditions which look like they might lie in G, based on information from T^G and finitely many α_i^G, which do lie in M_∞. Notice that M_∞ (in fact M_F for $F = \emptyset$) also contains α-good rank functions on T^G, for each $\alpha < \omega_1^{CK}$. To overcome the second obstacle, we use Lemma 6.30.

Fix $p \in G$ such that $p \Vdash \forall n(\psi(n) \vee \varphi(n))$ (note that $\forall n(\psi(n) \vee \varphi(n))$ is Σ-over-$\mathcal{L}_F$). By genericity and by expanding F if necessary, we may assume that $F = \mathrm{dom}(f^p)$. By Lemma 6.30, fix $\mu < \omega_1^{CK}$ large enough such that $p \Vdash \forall n(\psi^\mu(n) \vee \varphi^\mu(n))$ and μ is greater than the rank of any constant in φ and ψ. So $M_\infty \models \forall n(\psi^\mu(n) \vee \varphi^\mu(n))$. By upward persistence, $M_\infty \models \forall n(\psi^\mu(n) \leftrightarrow \neg\varphi^\mu(n))$.

Next, fix $\nu < \omega_1^{CK}$ large enough such that $p \in \mathbb{P}_\nu$ and $\mathrm{rk}(\varphi^\mu(\mathbf{n}) \vee \psi^\mu(\mathbf{n})) < \nu$ for all n. Now, we define D as follows: $d \in D$ if and only if there is $q \in \mathbb{P}_{\omega\nu+\omega^2+\omega}$ extending p such that

1. $q \Vdash \psi^\mu(\mathbf{d})$;
2. $T^q \subset T^G$;
3. h^q is $(\omega\nu + \omega^2 + \omega)$-good;
4. $\forall i \in F(f^q(i) = \alpha_i^G \cap T^q)$.

Observe that D is hyperarithmetic in $T^G \oplus \langle \alpha_i^G \rangle_{i \in F}$, so $D \in M_F \subseteq M_\infty$. We show that $M_\infty \models \psi(\mathbf{d})$ if and only if $d \in D$.

Suppose that $M_\infty \models \psi(\mathbf{d})$. Then $M_\infty \models \psi^\mu(\mathbf{d})$, so we may fix $q^* \in G$ extending p which forces $\psi^\mu(\mathbf{d})$. We retag q^* to $q \in \mathbb{P}_{\omega\nu+\omega^2+\omega}$, defined as follows: $T^q = T^{q^*}$, $f^q = f^{q^*}$, $l^q = l^{q^*}$, and

$$h^q(\sigma) = \begin{cases} \infty & \text{if } h^{q^*}(\sigma) \geq \omega\nu + \omega^2 + \omega \\ h^{q^*}(\sigma) & \text{otherwise} \end{cases}.$$

Since $h^{q^*} \subseteq h^G$, h^q is $(\omega\nu + \omega^2 + \omega)$-good. It is easy to see that q is a condition in $\mathbb{P}_{\omega\nu+\omega^2+\omega}$. Since $p \in \mathbb{P}_\nu$, $h^q \supseteq h^p$. It is then easy to see that $q \leq p$. Observe that $\mathrm{Ret}_\leq(\omega\nu + \omega^2, q^*, q)$ and $F = \mathrm{dom}(f^p) \subseteq \mathrm{dom}(f^q) = \mathrm{dom}(f^{q^*})$, so by Lemma 6.26, $q \Vdash \psi^\mu(\mathbf{d})$. q witnesses that $d \in D$, as desired.

On the other hand, suppose that $M_\infty \models \varphi(\mathbf{d})$. Then $M_\infty \models \varphi^\mu(\mathbf{d})$, so we may fix $r \in G$ extending p which forces $\varphi^\mu(\mathbf{d})$. Suppose towards a contradiction that $d \in D$, as witnessed by some q.

Without loss of generality, we may assume that $\mathrm{dom}(f^q) \cap \mathrm{dom}(f^r) = F$. This is arranged by fixing F and permuting $\mathrm{dom}(f^q) \setminus F$ as necessary. This permutation

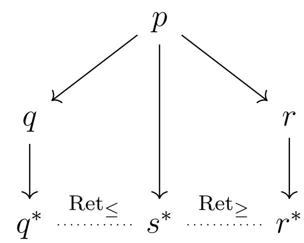

Figure 6.3: p and r lie in G, while q "looks like" it lies in G.

clearly preserves (2), (3), and (4) in the definition of D. By Lemma 6.28, (1) is also preserved.

Next, we aim to define $q^* \leq q$, $r^* \leq r$, and $s^* \leq p$ such that $\mathrm{Ret}_{\leq}(\omega\nu+\omega^2, q^*, s^*)$ and $\mathrm{Ret}_{\leq}(\omega\nu+\omega^2, r^*, s^*)$. This plan is illustrated in Figure 6.3. If we then apply Lemma 6.26 twice, we obtain that $s^* \Vdash \psi^\mu(\mathbf{d}) \wedge \varphi^\mu(\mathbf{d})$. But $s^* \leq p \in G$ and $M_\infty \models \forall n(\psi^\mu(n) \leftrightarrow \neg\varphi^\mu(n))$, giving the desired contradiction.

Define $T^{q^*} = T^{r^*} = T^{s^*} = T^r \cup T^q$. We define q^* as follows:

- $f^{q^*}(i) = \alpha_i^G \cap T^{q^*}$ for $i \in F$, and $f^{q^*}(i) = f^q(i)$ for $i \in \mathrm{dom}(f^q) \backslash F$. Also, for each n for which $\exists i(\langle n \rangle \subseteq f^r(i))$ but $\neg\exists i(\langle n \rangle \subseteq f^q(i))$, we add $\langle n \rangle$ to f^{q^*} (using numbers $> \mathrm{dom}(f^r)$ as domain);
- $l^{q^*} = l^q \cup l^r$;
- Define h^{q^*} by cases:

$$h^{q^*}(\tau) = \begin{cases} h^q(\tau) & \text{if } \tau \in T^q \\ \infty & \exists i(\tau \subseteq f^{q^*}(i)) \text{ or } |\tau| \leq 1 \\ h^G(\tau) & \text{if } h^G(\tau) < \omega\nu + \omega^2 \\ \omega\nu + \omega^2 + |\tau|_Q & \text{otherwise} \end{cases}$$

where $Q = \{\tau \in T^{q^*} : h^G(\tau) \geq \omega\nu + \omega^2\}$.

h^{q^*} **is well-defined.** First we show that the second and third case are mutually exclusive. Suppose $\exists i(\tau \subseteq f^{q^*}(i))$. There are three cases. If $i \in F$, then $\tau \subset \alpha_i^G$, so $h^G(\tau) = \infty > \omega\nu + \omega^2$. If $i \in \mathrm{dom}(f^q)\backslash F$, then $f^{q^*}(i) = f^q(i)$, so $h^q(\tau) = \infty$. Since q is $(\omega\nu + \omega^2 + \omega)$-good, $h^G(\tau) \geq \omega\nu + \omega^2 + \omega > \omega\nu + \omega^2$. In the remaining case, $|f^{q^*}(i)| = 1$ so $h^G(\tau) = \infty > \omega\nu + \omega^2$.

Next, the first and third case do not conflict: if $\tau \in T^q$ and $h^G(\tau) < \omega\nu + \omega^2$, then $h^q(\tau) = h^G(\tau)$ because h^q is $(\omega\nu + \omega^2 + \omega)$-good.

Finally, we show that the first and second case do not conflict. If $\tau \in T^q$ and $\exists i(\tau \subseteq f^{q^*}(i))$, we consider three cases. If $i \in \mathrm{dom}(f^q)\backslash F$, then $h^q(\tau) = \infty$ as before. If $i \in F$, then $f^{q^*}(i) = \alpha_i^G \cap T^{q^*}$. Since $\tau \in T^q$, it follows that $\tau \subseteq \alpha_i^G \cap T^q = f^q(i)$, so $h^q(\tau) = \infty$. Otherwise, $|f^{q^*}(i)| = 1$ so $h^q(\tau) = \infty$. We have shown that h^{q^*} is well-defined.

q^* **is a condition.** It suffices to check that h^{q^*} is a rank function. The nontrivial cases are as follows:

<u>Case 1.</u> $|\tau| > 1$, $\tau \notin T^q$, $\tau^- \in T^q$, $h^G(\tau) < \omega\nu + \omega^2$. We show that $h^{q^*}(\tau) < h^{q^*}(\tau^-)$, i.e., $h^G(\tau) < h^q(\tau^-)$. If $h^q(\tau^-) \geq \omega\nu + \omega^2$, then we are done. Otherwise, $h^q(\tau^-) < \omega\nu + \omega^2$, so $h^G(\tau^-) = h^q(\tau^-)$ since q is $(\omega\nu + \omega^2 + \omega)$-good. So $h^G(\tau) < h^G(\tau^-) = h^q(\tau^-)$ as desired.

<u>Case 2.</u> $|\tau| > 1$, $\tau \notin T^q$, $\tau^- \in T^q$, $h^G(\tau) \geq \omega\nu + \omega^2$, $h^{q^*}(\tau) = \omega\nu + \omega^2 + |\tau|_Q$.

If $h^{q^*}(\tau^-) \geq \omega\nu + \omega^2 + \omega$, then $h^{q^*}(\tau^-) \geq \omega\nu + \omega^2 + \omega > \omega\nu + \omega^2 + |\tau|_Q = h^{q^*}(\tau)$ as desired.

Otherwise, $h^{q^*}(\tau^-) < \omega\nu + \omega^2 + \omega$. Then

$$\begin{aligned} h^{q^*}(\tau^-) &= h^q(\tau^-) && \tau^- \in T^q \\ &= h^G(\tau^-) && q \text{ is } (\omega\nu+\omega^2+\omega)\text{-good} \\ &> h^G(\tau) && h^G \text{ is a rank function} \\ &\geq \omega\nu + \omega^2 + |\tau|_Q && \text{Proposition 6.12} \\ &= h^{q^*}(\tau) \end{aligned}$$

as desired. This shows that h^{q^*} is a rank function.

q^* **extends** q**.** (5), (7), and (8) hold by definition. (6b) holds because all new paths we added (if any) have length 1. (9) holds because for any new path $\langle m \rangle$ that we added,

$$m \notin \{n : \exists i(\langle n \rangle \subseteq f^q(i))\}$$

so we are free to add $\langle m \rangle$ to q.

It remains to check (6a): for $i \in \mathrm{dom}(f^q) \backslash F$, $f^{q^*}(i) = f^q(i)$ so we are done. As for $i \in F$, since q witnesses that $d \in D$, $f^{q^*}(i) \cap T^q = \alpha_i^G \cap T^{q^*} \cap T^q = \alpha_i^G \cap T^q = f^q(i)$ as desired.

Next we define r^*:

- $f^{r^*} = \alpha_i^G \cap T^{r^*}$ for $i \in F$, and $f^{r^*}(i) = f^r(i)$ for $i \in \mathrm{dom}(f^r) \backslash F$. Also, for each n for which $\exists i(\langle n \rangle \subseteq f^q(i))$ but $\neg\exists i(\langle n \rangle \subseteq f^r(i))$, we add $\langle n \rangle$ to f^{r^*} (using numbers $> \mathrm{dom}(f^{q^*})$ as domain);
- $l^{r^*} = l^q \cup l^r$;
- $h^{r^*} = h^G \restriction T^{r^*}$.

r^* **is a condition.** It suffices to show that for each $i \in \mathrm{dom}(f^{r^*})$, $h^{r^*}(f^{r^*}(i)) = \infty$. For $i \in \mathrm{dom}(f^r)\backslash F$, $h^{r^*}(f^{r^*}(i)) = h^G(f^r(i)) = \infty$ because $r \in G$. For $i \in F$, $h^{r^*}(f^{r^*}(i)) = h^G(\alpha_i^G \cap T^{r^*}) = \infty$. Otherwise, $|f^{r^*}(i)| = 1$ so $h^{r^*}(f^{r^*}(i)) = \infty$.

r^* **extends** r**.** (5) and (8) hold by definition. (7) holds because $r \in G$. (6b) holds because all new paths we added (if any) have length 1. (9) holds because for any new path $\langle m \rangle$ that we added,

$$m \notin \{n : \exists i(\langle n \rangle \subseteq f^r(i))\}$$

so we are free to add $\langle m \rangle$ to r.

It remains to check (6a): for $i \in \mathrm{dom}(f^r)\backslash F$, $f^{r^*}(i) = f^r(i)$ so we are done. As for $i \in F$, $f^{r^*}(i) \cap T^r = \alpha_i^G \cap T^{r^*} \cap T^r = \alpha_i^G \cap T^r = f^r(i)$ because $r \in G$. We have showed that r^* is a condition extending r.

Finally, define s^* as follows:

- $f^{s^*} = f^{q^*} \cup f^{r^*}$;
- $l^{s^*} = l^q \cup l^r$;
- Define h^{s^*} by cases:

$$h^{s^*}(\tau) = \begin{cases} h^G(\tau) & \text{if } h^G(\tau) < \omega\nu + \omega^2 \\ \infty & \text{if } h^G(\tau) \geq \omega\nu + \omega^2 \end{cases}.$$

f^{s^*} **is well-defined.** This holds because $\mathrm{dom}(f^{q^*}) \cap \mathrm{dom}(f^{r^*}) = F$ and $f^{q^*} \restriction F = f^{r^*} \restriction F$.

s^* **is a condition.** It suffices to show that if $i \in \mathrm{dom}(f^{s^*})$, then $h^{s^*}(f^{s^*}(i)) = \infty$. First suppose that $i \in \mathrm{dom}(f^{q^*})$. It suffices to show that $h^G(f^{q^*}(i)) \geq \omega\nu + \omega^2$.

If not, then by definition of h^{q^*}, $h^{q^*}(f^{q^*}(i)) = h^G(f^{q^*}(i))$. But the former is ∞ because q^* is a condition, contradicting our assumption. Hence $h^G(f^{q^*}(i)) \geq \omega\nu + \omega^2$. By definition of h^{s^*}, $h^{s^*}(f^{q^*}(i)) = \infty$ as desired.

Otherwise, $i \in \mathrm{dom}(f^{r^*})$. Then $h^G(f^{r^*}(i)) = h^{r^*}(f^{r^*}(i)) = \infty \geq \omega\nu + \omega^2$, so $h^{s^*}(f^{r^*}(i)) = \infty$ as desired.

s^* **extends** p**.** (5) and (8) hold by definition. (6a), (6b), and (9) hold because both $q^* \leq p$ and $r^* \leq p$ satisfy (6a), (6b), and (9). (7) holds because $p \in \mathbb{P}_\nu$, $p \in G$, and h^{s^*} is $(\omega\nu + \omega^2)$-good.

q^***,** r^* **and** s^* **satisfy** $\mathrm{Ret}_{\leq}(\omega\nu + \omega^2, q^*, s^*)$ **and** $\mathrm{Ret}_{\leq}(\omega\nu + \omega^2, r^*, s^*)$**.** First, by definition of h^{q^*}, h^{r^*}, and h^{s^*}, they are $(\omega\nu + \omega^2)$-good. Also, by definition, $f^{q^*} = f^{s^*} \upharpoonright \mathrm{dom}(f^{q^*})$ and $f^{r^*} = f^{s^*} \upharpoonright \mathrm{dom}(f^{r^*})$.

Second, by construction, $\{n : \exists i(\langle n \rangle \subseteq f^{q^*}(i))\}$ and $\{n : \exists i(\langle n \rangle \subseteq f^{r^*}(i))\}$ (and hence their union $\{n : \exists i(\langle n \rangle \subseteq f^{s^*}(i))\}$) are all equal to

$$\{n : \exists i(\langle n \rangle \subseteq f^q(i))\} \cup \{n : \exists i(\langle n \rangle \subseteq f^r(i))\}.$$

Finally, $l^{s^*} = l^{q^*} = l^{r^*}$ by definition. $\square$

CHAPTER 7

HALIN'S THEOREM ON DISJOINT RAYS

This chapter is part of joint work with James Barnes and Richard A. Shore.

In this chapter, all graphs are undirected unless specified otherwise. A *path* in a graph is a (possibly finite) sequence of distinct vertices such that consecutive vertices are adjacent. A *ray* is an infinite path (indexed by $\mathbb{N}$). A set of paths or rays is *vertex-disjoint* (*edge-disjoint*) if the rays within do not have any vertices (edges, respectively) in common.

In 1965, Halin [21] proved the following:

Theorem 7.1. *In every graph, there is a set of vertex-disjoint rays of maximum cardinality. In particular, if a graph contains k many vertex-disjoint rays for every $k \in \mathbb{N}$, then it contains a set of infinitely many vertex-disjoint rays.*

Halin [21] also proved the analogous result for edge-disjoint rays, but we will not study it here. Henceforth, we will write disjoint instead of vertex-disjoint.

In this chapter, we study the second statement in Halin's theorem (restricted to countable graphs) from the point of view of reverse mathematics. Henceforth we will simply refer to this statement as Halin's theorem.

First, observe that we cannot collect multiple disjoint rays in a greedy manner. For example, in the $\mathbb{N} \times \mathbb{N}$ grid, there are rays which pass through every vertex, and hence are not part of any set of infinitely many disjoint rays. This suggests that Halin's theorem has nontrivial computational and proof-theoretic content, as we will show in Proposition 7.8.

Next, notice that the form of Halin's theorem is reminiscent of a compactness theorem. However, rays are not first-order objects. In fact, they can be hard to compute! Even if we can compute k-many disjoint rays individually, we cannot in general compute a set of k-many disjoint rays uniformly in k (see Proposition 7.15).

This suggests the following two formalizations of Halin's theorem in the language of second-order arithmetic. The only difference between them lies in their hypotheses, about which sets are asserted to exist.

Definition 7.2. Define the *infinite ray theorem* (IRT) and the *weak infinite ray theorem* (WIRT):

IRT: if G is a graph and for each k, there is a set of k disjoint G-rays, then there is a set of infinitely many disjoint G-rays.

WIRT: if G is a graph and there is a sequence of sets $(X_k)_k$ such that for each $k \in \mathbb{N}$, X_k is a set of k disjoint G-rays, then there is a set of infinitely many disjoint G-rays.

Trivially, IRT implies WIRT. We will show that WIRT is much weaker than IRT. In more detail, our main results are as follows:

- WIRT is provable in ACA_0 (Theorem 7.5) but not RCA_0 (Proposition 7.8);
- IRT is provable in Σ^1_1-AC_0 (Proposition 7.14) and implies ABW_0 over $\mathsf{RCA}_0 + I\Sigma^1_1$ (Proposition 7.16).

Our results imply that IRT is a theorem of hyperarithmetic analysis (Definition 6.4). To our knowledge, IRT is only the second known theorem of hyperarithmetic analysis which is "natural", i.e., formulated without concepts from logic. (Montalbán [30] discovered the first such theorem; see Theorem 6.9 and the paragraph after it.)

7.1 The weak infinite ray theorem

In this section, we give some upper and lower bounds on WIRT and its variants. We begin with some preliminaries.

If R is a ray and x is a vertex on R, let Rx denote the initial segment of R up until x. Let xR denote the tail of R starting from x. If x precedes y on R, let xRy denote the path starting from x, following R, and ending at y. Analogously, if x precedes y on R and y precedes z on S, let $xRySz$ denote the path which starts from x, then follows R until y, and then follows S until z.

An important tool in the proof of Halin's theorem is Menger's theorem for finite graphs (see [14, Theorem 3.3.1]). If A and B are disjoint sets of vertices in a graph, we say that P is an *A-B path* if P starts with some vertex in A and ends with some vertex in B. A set of vertices S *separates A and B* if any A-B path passes through at least one vertex in S.

Theorem 7.3 (Menger)**.** *Let G be a finite graph. If A and B are disjoint sets of vertices in G, then the minimum size of a set of vertices which separate A and B is equal to the maximum size of a set of disjoint A-B paths.*

7.1.1 Upper bounds

Diestel [14, Theorem 8.2.5] presents a proof of Halin's theorem due to Andreae. The key combinatorial lemma implicit in Andreae's proof is as follows:

> Given a set of n disjoint rays $R_0, \ldots, R_{n-1}$ and a set of n^2+1 disjoint rays $S_0, \ldots, S_{n^2}$, there is a set of $n+1$ disjoint rays $R'_0, \ldots, R'_n$ such that for each $i < n$, R_i and R'_i start at the same vertex.

On the face of it, constructing such $R'_0, \ldots, R'_n$ could be difficult; perhaps as difficult as providing a solution to a Σ^1_1 predicate. However, Andreae's proof actually constructs $R'_0, \ldots, R'_n$ such that for each $i \leq n$, R'_i shares a tail with some R_j or S_j. This lowers the complexity of constructing such rays considerably, allowing us to prove the following effective version:

Theorem 7.4. *If G is a graph and $(X_k)_k$ is such that for each $k \in \mathbb{N}$, X_k is a set of k disjoint rays, then $G \oplus ((X_k)_k)'$ uniformly computes a set of infinitely many disjoint rays.*

Proof. For later purposes, we fix the polynomial $f(n) = \frac{n(n+1)}{2} + n^2 + 1$. (Any computable function which majorizes f will do.)

Fix a graph G and a sequence $((S^k_i)_{i<k})_k$ such that for each k, $\{S^k_i : i < k\}$ is a set of disjoint G-rays. We construct an infinite set of disjoint G-rays by recursion. Start with the empty set of rays.

At the beginning of stage n, we will have constructed n many disjoint rays $R^n_0, \ldots, R^n_{n-1}$, where each R^n_i shares a tail with some $S^{f(k)}_j$, $j < f(k)$, $k < n$ (and hence can be coded by a number, relative to $((S^k_i)_{i<k})_k$.) For each $i < n$, let x^n_i be the $(n-i)^{\text{th}}$ vertex on R_i^n. The path $R_i^n x_i^n$ will be an initial segment of the i^{th} ray in our eventual set of infinitely many disjoint rays.

In the following, we construct $n+1$ many disjoint rays $R_0^{n+1}, \ldots, R_n^{n+1}$ such that

- each R_i^{n+1} shares a tail with some $S_j^{f(k)}$, $j < f(k)$, $k < n+1$;
- for each $i < n$, $R_i^{n+1} x_i^n = R_i^n x_i^n$. (This ensures that $(\lim_n R_i^n)_i$ exists, consists of disjoint rays, and is computable from $((R_i^n)_{i<n})_n$.)

First, use $(((S_i^k)_{i<k})_k)'$ to compute the set

$$Q_0 = \{q < f(n) : S_q^{f(n)} \text{ is disjoint from } R_i^n x_i^n \text{ for } i < n\}.$$

Note that

$$|Q_0| \geq f(n) - \sum_{i<n} |R_i^n x_i^n| = f(n) - \sum_{i<n} (n-i) = n^2 + 1.$$

Next, we claim that there is a set $I \subseteq \{0, \ldots, n-1\}$ such that if we define

$$Q_1 = \{q \in Q_0 : S_q^{f(n)} \text{ meets } R_i^n \text{ for some } i \in I\}$$
$$\mathcal{R} = \{S_q^{f(n)} : q \in Q_1\},$$

then

- for each $i \in I$, R_i^n meets at least n rays in $\mathcal{R}$;
- for each $i < n$ outside I, R_i^n does not meet any ray in $\mathcal{R}$;
- $|Q_1| \geq |I|^2 + 1$.

Proof. First, use $(((S_i^k)_{i<k})_k)'$ to compute $\{(i, q) : i < n, q \in Q_0, R_i^n \text{ intersects } S_q^{f(n)}\}$. Then we construct I by recursion. Start with $I = \{0, \ldots, n-1\}$. While there is

$i \in I$ such that R_i^n meets less than n rays in $\mathcal{R}$, we remove i from I. (Notice that this may cause Q_1 and $\mathcal{R}$ to shrink by as much as $n-1$.) This process eventually terminates. We have that

$$|Q_1| \geq |Q_0| - n(n - |I|) \geq n^2 + 1 - n(n - |I|) = n|I| + 1 \geq |I|^2 + 1$$

as desired. □

$(((S_i^k)_{i<k})_k)'$ can uniformly construct I as above. For each $i < n$ outside I, we define

$$R_i^{n+1} = R_i^n.$$

Next, we define R_{n+1}^{n+1} and finally R_i^{n+1} for $i \in I$. Let m be the size of I. For each $i \in I$, use $(((S_i^k)_{i<k})_k)'$ to compute the first vertex z_i on R_i^n such that $R_i^n z_i$ (equivalently, $x_i^n R_i^n z_i$) meets m many rays in $\mathcal{R}$. We define

$$Z = \bigcup_{i \in I} x_i^n R_i^n z_i$$

$$Q_2 = \{q \in Q_1 : S_q^{f(n)} \text{ meets } Z\}.$$

Note that $|Q_2| \leq m^2$, so $Q_1 \backslash Q_2$ is nonempty. We then define

$$R_{n+1}^{n+1} = S_{\min(Q_1 \backslash Q_2)}^{f(n)}.$$

Finally, we define R_i^{n+1} for $i \in I$. For each $q \in Q_2$, use $(((S_i^k)_{i<k})_k)'$ to compute the first vertex y_q on $S_q^{f(n)}$ such that $y_q S_q^{f(n)}$ is disjoint from Z. We define

$$X = \{x_i^n : i \in I\}$$

$$Y = \{y_q : q \in Q_2\}$$

$$H = Z \cup \bigcup_{q \in Q_2} S_q^{f(n)} y_q.$$

We apply Menger's theorem to $X, Y \subseteq H$. We claim that X cannot be separated from Y in H by fewer than m vertices.

Suppose we have a set A of less than m vertices in H. Since $|I| = m$ and $\{R_i^n : i \in I\}$ is disjoint, there is some $i \in I$ such that R_i^n does not meet A. Also, since $x_i^n R_i^n z_i$ meets m many disjoint rays in $\{S_q^{f(n)} : q \in Q_2\}$, there is some $q \in Q_2$ such that $S_q^{f(n)}$ meets $x_i^n R_i^n z_i$ (say at z) but not A. Then $x_i^n R_i^n z S_q^{f(n)} y_q$ is a path in H from x_i^n to y_q which does not meet A. This proves our claim.

By Menger's theorem (Theorem 7.3), there is a set of m disjoint X-Y paths in H. We can computably search for such paths. Then, for each $i \in I$, define R_i^{n+1} by first following R_i^n up until x_i^n, then following the X-Y path given by Menger's theorem, and finally following whichever $S_q^{f(n)}$, $q \in Q_2$ that we are on. This completes stage n of the construction.

Then $(\lim_n R_i^n)_i$ is a set of infinitely many disjoint rays, as desired. □

The above proof shows that

Theorem 7.5. ACA_0 *implies* WIRT.

We can obtain better upper bounds for WIRT for restricted classes of graphs.

Proposition 7.6. RCA_0 *implies* WIRT *for trees.*

Proof. In the proof of Theorem 7.4, the only noneffective steps involved computing the intersection relations between sets of rays or paths. In a tree, such relations are uniformly Δ_1^0-definable. □

Next, we study lower bounds for WIRT. First we will construct a computable instance of WIRT with no computable solution, i.e., a computable graph G and a computable sequence of sets of k-many rays $(X_k)_k$ in G, such that no set of infinitely many disjoint rays is computable. Our results can be phrased in terms of computably enumerable equivalence relations.

7.1.2 Lower bounds via computably enumerable equivalence relations

An equivalence relation E on $\mathbb{N}$ is *computably enumerable* if the set $\{(x, y) : xEy\}$ is computably enumerable. Such an equivalence relation is known as a *ceer.*

Given a ceer E (specifically, an index e such that $W_e = \{(x, y) : xEy\}$), we can construct a computable graph G on $\mathbb{N} \times \mathbb{N}$ in the following way:

- (y, s) and $(y, s + 1)$ are adjacent if and only if for all $x < y$, $(x, y) \notin W_{e,s}$;
- (y, s) and $(x, s+1)$ are adjacent if and only if $x < y$ and $(x, y) \in W_{e,s+1} \setminus W_{e,s}$;
- no other vertices are adjacent.

We make some observations about G:

- G is highly recursive.
- Any ray in G must "grow upward", i.e., the second coordinates of its vertices must increase as one traverses the ray.
- If R is a ray, then $\{x : \exists s (x, s) \in R\}$ is contained in an E-class.
- Two rays R_0 and R_1 are disjoint if and only if $\{x : \exists s (x, s) \in R_0\}$ and $\{x : \exists s (x, s) \in R_1\}$ are in different E-classes.

– If I is an independent set in E, then one can uniformly compute a set of $|I|$ many disjoint rays: for each $x \in I$, start at $(x, 0)$ and grow upwards.

Therefore, disjoint rays in G are closely connected to independent sets in E. In this section, we will construct ceers with various properties, and then apply the above graph construction to derive various results in reverse mathematics.

Proposition 7.7. *There is a ceer E such that one can uniformly in n compute an E-independent set of size n, yet there is no infinite c.e. E-independent set.*[1]

Proof. We will construct a ceer E on $\{\langle n, m\rangle : n < m\}$ such that:

– for each m, the m^{th} *block* $\{\langle n, m\rangle : n < m\}$ is E-independent;

– there is no infinite c.e. E-independent subset of $\{\langle n, m\rangle : n < m\}$.

(We only defined ceers with field $\mathbb{N}$, so at the end of our construction we can use a computable bijection between $\{\langle n, m\rangle : n < m\}$ and $\mathbb{N}$ to change the field of E to $\mathbb{N}$.)

To satisfy the second condition above, we have a requirement R_e for each index e stating that Φ_e does not enumerate an infinite E-independent set. We arrange the requirements in order of priority: R_d has higher priority than R_e if $d < e$. At any stage, each requirement is in one of three states: satisfied, unsatisfied with witness, and unsatisfied without witness. At the beginning of our construction, all requirements are unsatisfied without witness.

At stage s of the construction, we run one more step of each Φ_e, $e < s$. If any of the conditions below hold, we proceed accordingly.

[1]The ceers for which there is no infinite c.e. independent set are known as dark ceers. Andrews and Sorbi [3] proved many structural results about dark ceers.

1. If Φ_e enumerates some number outside $\{\langle n,m\rangle : n < m\}$, then R_e releases all restraints and is (forever) satisfied.

2. If R_e is unsatisfied with witness, and Φ_e enumerates some $\langle n',m'\rangle$ which lies in $[\langle n,m\rangle]_{E_s}$, then R_e releases all restraints and is (forever) satisfied.

3. If R_e is unsatisfied without witness and Φ_e enumerates some $\langle n,m\rangle$ which is not restrained by requirements of higher priority, then R_e chooses $\langle n,m\rangle$ as its witness. Furthermore, for all $\langle n',m'\rangle$ in $[\langle n,m\rangle]_{E_s}$ (including $\langle n,m\rangle$ itself), R_e restrains all equivalence classes $[\langle i,m'\rangle]_{E_s}$, $i \neq n'$. This means that no requirements of priority e or lower are allowed to act by expanding any of these equivalence classes.

4. If R_e is unsatisfied with witness $\langle n,m\rangle$, and Φ_e enumerates some $\langle n',m'\rangle$ which is not restrained by requirements of priority e or higher, then we act to satisfy R_e by adding $\langle n',m'\rangle$ to $[\langle n,m\rangle]_{E_{s+1}}$. R_e releases all restraints and is now (forever) satisfied. Furthermore, any requirements of lower priority which were unsatisfied with witness now become unsatisfied without witness. (Any requirements of lower priority which were satisfied remain satisfied.)

This completes stage s of the construction.

We verify that E has the desired properties. Each block is E-independent because we only enumerate equivalences into E in case 4, and such action must respect the restraints of the relevant R_e.

Next, we show that no Φ_e enumerates an infinite E-independent subset of $\{\langle n,m\rangle : n < m\}$. Go to a sufficiently late stage s such that all R_d, $d < e$ which will ever act have already acted. At this stage, the R_d for $d < e$ only restrain finitely many classes, which will never expand from now on. Therefore if Φ_e enumerates an infinite subset of $\{\langle n, m\rangle : n < m\}$, R_e will eventually choose a witness (if it has not already) and we will eventually act to satisfy R_e. $\square$

The above proposition shows that

Proposition 7.8. WIRT *restricted to highly recursive graphs is not provable in* RCA_0.

Note that the restriction of WIRT to graphs constructed from ceers using the above method cannot imply ACA_0:

Proposition 7.9. *Let E be a ceer such that one can uniformly in n compute an E-independent set of size n. Then there is a nonempty Π^0_1 class of infinite E-independent sets. Therefore (by, e.g., the low basis theorem) there is an infinite E-independent set which does not compute $\emptyset'$.*

Proof. Fix a computable sequence $(X_n)_n$ such that for each n, X_n is (the code for) an E-independent set of size n. Consider the following computably branching subtree T of $\mathbb{N}^{<\mathbb{N}}$: $\sigma \in T$ if and only if

- for each $n < |\sigma|$, $\sigma(n) \in X_n$;
- $\{\sigma(n) : n < |\sigma|\}$ is $E_{|\sigma|}$-independent. ($E_{|\sigma|}$ is the enumeration of E up until stage $|\sigma|$.)

It is easy to see that T is infinite and that for any path P on T, $\{P(n) : n \in \mathbb{N}\}$ is an infinite E-independent set. $\square$

We do not know if WIRT implies ACA_0.

On the other hand, if we remove the restriction that one can uniformly in n compute an E-independent set of size n, then we can code $\emptyset'$ into the infinite E-independent sets (and more).

Proposition 7.10. *There is a ceer E with infinitely many classes such that if $(X_i)_i$ is a sequence such that for each i, X_i is (the code for) an E-independent set of size i, then $(X_i)_i$ uniformly computes $\emptyset'$. In particular, every infinite E-independent set uniformly computes $\emptyset'$.*

Proof. We will compute $\emptyset'$ by majorizing its true stage function ∇, defined as follows. Fix an enumeration $(\emptyset'_s)_s$ of $\emptyset'$. For each i, $\nabla(i)$ is the least stage s greater than $\nabla(i-1)$ such that $\emptyset'_s \upharpoonright i = \emptyset' \upharpoonright i$. (By convention, $\nabla(-1) = -1$.) Any function that majorizes ∇ uniformly computes $\emptyset'$.

Consider the following equivalence relation E: $n_0 E n_1$ if and only if there is some i such that $n_0, n_1 \in (\nabla(i-1), \nabla(i)]$. If $(X_i)_i$ is a sequence such that for each i, X_i is (the code for) an E-independent set of size i, then the function $i \mapsto \max(X_{i+1})$ majorizes ∇ and hence uniformly computes $\emptyset'$.

It remains to show that E can be computably enumerated. At stage t, we can guess $\nabla(j)$ for $j < t$ as follows. For each $j < t$, define $\nabla_t(j)$ to be the least stage $s \leq t$ greater than $\nabla_t(j-1)$ such that $\emptyset'_s \upharpoonright j = \emptyset'_t \upharpoonright j$. (By convention, $\nabla_t(-1) = -1$.) Define E_t to be the equivalence relation with the following classes:

- $(\nabla_t(j-1), \nabla_t(j)]$ for $j < t$;
- $\{n\}$ for $n > \nabla_t(t-1)$.

It is easy to see that E_t is uniformly computable in t, and $E = \bigcup_t E_t$. □

The above proof shows that

Proposition 7.11. *The principle "if G is a graph and for each k, there is a set of k disjoint G-rays, then there is a sequence of sets $(X_k)_k$ such that for each k, X_k is a set of k disjoint G-rays" implies* ACA_0.

We will show in Proposition 7.15 that the above principle is actually equivalent to IRT, which is much stronger than ACA_0 (Proposition 7.16).

Proposition 7.10 also yields a reversal to ACA_0 from the following adhoc (apparent) strengthening of WIRT.

Definition 7.12. Define *nonuniform*-WIRT as follows: if G is a graph and there is a sequence of G-rays $(R_i)_i$ such that for each k, there are $i_1, \ldots, i_k$ such that $R_{i_1}, \ldots, R_{i_k}$ are disjoint G-rays, then there is a set of infinitely many disjoint G-rays.

Proposition 7.13. *Nonuniform*-WIRT *is equivalent to* ACA_0 *over* RCA_0.

Proof. The proof of Proposition 7.10 shows that nonuniform-WIRT implies ACA_0. It remains to show that nonuniform-WIRT is provable in ACA_0. Suppose we have a graph G and a sequence of rays $(R_i)_i$ such that for each k, there are $i_1, \ldots, i_k$ such that $R_{i_1}, \ldots, R_{i_k}$ are disjoint rays. Then ACA_0 can choose, for each k, the least $i_1, \ldots, i_k$ such that $R_{i_1}, \ldots, R_{i_k}$ are disjoint. This yields a sequence of sets of rays $(X_k)_k$ such that for each k, X_k is a set of k disjoint rays. Finally, we apply WIRT (which is provable in ACA_0 by Theorem 7.5) to produce a set of infinitely many disjoint rays as desired. □

7.2 The infinite ray theorem

We now turn our attention to IRT. We show that IRT is a theorem of hyperarithmetic analysis and study its relationships with other theories of hyperarithmetic analysis.

Instead of verifying directly that IRT satisfies the definition of a theorem of hyperarithmetic analysis, we will show that IRT lies between two known theories of hyperarithmetic analysis: Σ_1^1-AC_0 and ABW_0.

Proposition 7.14. Σ_1^1-AC_0 *implies* IRT.

Proof. Σ_1^1-AC_0 proves that the assumption in IRT implies the assumption in WIRT. Since ACA_0 proves WIRT (Theorem 7.5), the desired result follows. □

Before giving lower bounds for IRT, we digress slightly:

Proposition 7.15. *The principle "if G is a graph and for each k, there is a set of k disjoint G-rays, then there is a sequence of sets $(X_k)_k$ such that for each k, X_k is a set of k disjoint G-rays" is equivalent to* IRT.

Proof. Clearly, the given principle follows from IRT. Conversely, we showed in Proposition 7.11 that the given principle implies ACA_0 and hence WIRT. Together with WIRT, it implies IRT as desired. □

Returning to lower bounds for IRT, we begin by noting that IRT implies ACA_0 (as we just showed, using Proposition 7.11). We show below that IRT implies ABW_0 over $\mathsf{RCA}_0 + I\Sigma_1^1$. This strengthens Conidis's result that Σ_1^1-AC_0 implies ABW_0 over $\mathsf{RCA}_0 + I\Sigma_1^1$.

Proposition 7.16 ($I\Sigma^1_1$)**.** IRT *implies* ABW_0.

Proof. Suppose $A(X)$ is an arithmetic predicate on $2^{\mathbb{N}}$ which does not have finitely many solutions. By $I\Sigma^1_1$, for each n, there is some set of n-many distinct solutions of $A(X)$. (This is the only use of $I\Sigma^1_1$ in this proof.)

Next, by a lemma of Simpson [42, V.5.4], there is a computable tree T such that ACA_0 proves that

$$\forall X(A(X) \leftrightarrow \exists f((X,f) \in [T]) \quad \text{and} \quad \forall X(\exists \text{ at most one } f)((X,f) \in [T]).$$

We show that T is an instance of IRT. If $A(X)$ holds, then by ACA_0, there is some f such that $(X,f) \in [T]$. Therefore for each n, T (as a tree) has at least n-many paths. By taking an appropriate tail of each path, it follows that T (as a graph) has at least n-many disjoint rays.

Now, apply IRT to T to obtain an infinite sequence of disjoint rays in T. By extending or truncating each of those rays to the root of T, we obtain an infinite sequence of distinct paths in T, say $(X_n, f_n)_n$. Since for each X, there is at most one f such that $(X,f) \in [T]$, it follows that $(X_n)_n$ is an infinite sequence of distinct solutions of A.

Next, we follow a well-known proof of the Bolzano-Weierstrass theorem from König's lemma. Using ACA_0, define the tree

$$T = \{\sigma \in 2^{<\mathbb{N}} : \exists^\infty n(\sigma \prec X_n)\}.$$

Using $I\Sigma_2$, we can show that each level of T is nonempty. By König's lemma, we obtain a path Z on T. It is easy to see that Z is an accumulation point of $\{X_n : n \in \mathbb{N}\}$, and hence an accumulation point of $\{X : A(X)\}$. $\square$

Propositions 7.14 and 7.16 imply that

Theorem 7.17. IRT *is a theorem of hyperarithmetic analysis.*

Next, we discuss separations between IRT and other theories of hyperarithmetic analysis. One model of interest is van Wesep's model $\mathcal{N}$. van Wesep [44, Theorem 1.1] constructed $\mathcal{N}$ and showed that it satisfies unique-Σ^1_1-AC but not Δ^1_1-CA. Conidis [13, section 4] strengthened van Wesep's result to show that $\mathcal{N}$ satisfies ABW. (Neeman [32, Theorem 1.3] also strengthened van Wesep's result to show that $\mathcal{N}$ does not satisfy INDEC, but we will not need that here.)

We observe that $\mathcal{N}$ fails to satisfy IRT:

Theorem 7.18. *There is an ω-model satisfying* ABW *but not* IRT*. Therefore* ABW *does not imply* IRT*.*

Proof. Consider van Wesep's $\mathcal{N}$. Conidis [13, section 4] showed that $\mathcal{N}$ satisfies ABW. We claim that $\mathcal{N}$ fails to satisfy IRT. Let T^G be the generic tree that was constructed in the construction of $\mathcal{N}$. Since $\mathcal{N}$ contains infinitely many paths on T^G, $\mathcal{N}$ thinks that T^G is an instance of IRT. But $\mathcal{N}$ does not contain any infinite sequence of paths on T^G. (This fact was used in the proof of Lemma 1.4 of van Wesep [44]. It was essentially proved by Steel [43, Lemma 7].) So $\mathcal{N}$ does not contain any IRT-solution to T^G. □

Our results are summarized in Figure 7.1. To simplify the diagram, we use the base theory $\mathsf{RCA}_0 + I\Sigma^1_1$ instead of RCA_0.

We end with some open questions regarding the above theories of hyperarithmetic analysis:

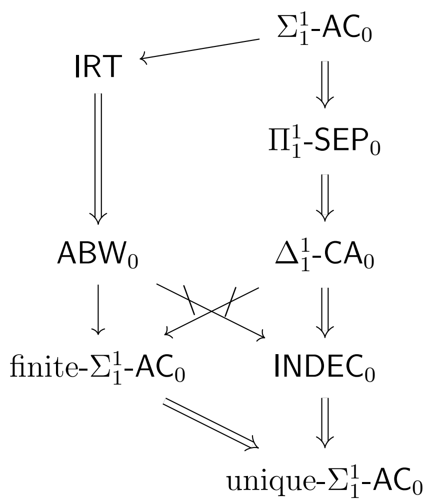

Figure 7.1: Partial zoo of theories of hyp analysis (assuming $I\Sigma^1_1$)

1. Does IRT imply Σ^1_1-AC_0, INDEC_0, or any theory in between?

2. Does Π^1_1-SEP_0 imply IRT? Does Π^1_1-SEP_0 imply ABW_0?

BIBLIOGRAPHY

[1] Ron Aharoni. König's duality theorem for infinite bipartite graphs. *J. London Math. Soc. (2)*, 29(1):1–12, 1984.

[2] Ron Aharoni, Menachem Magidor, and Richard A. Shore. On the strength of König's duality theorem for infinite bipartite graphs. *J. Combin. Theory Ser. B*, 54(2):257–290, 1992.

[3] Uri Andrews and Andrea Sorbi. Joins and meets in the structure of ceers. https://arxiv.org/abs/1802.09249, 2018.

[4] Dwight R. Bean. Effective coloration. *J. Symbolic Logic*, 41(2):469–480, 1976.

[5] Vasco Brattka, Matthew de Brecht, and Arno Pauly. Closed choice and a uniform low basis theorem. *Ann. Pure Appl. Logic*, 163(8):986–1008, 2012.

[6] Vasco Brattka and Guido Gherardi. Weihrauch degrees, omniscience principles and weak computability. *J. Symbolic Logic*, 76(1):143–176, 2011.

[7] Vasco Brattka, Guido Gherardi, and Alberto Marcone. The Bolzano-Weierstrass theorem is the jump of weak König's lemma. *Ann. Pure Appl. Logic*, 163(6):623–655, 2012.

[8] Vasco Brattka, Guido Gherardi, and Arno Pauly. Weihrauch complexity in computable analysis. arXiv:1707.03202, 2018.

[9] Vasco Brattka and Arno Pauly. On the algebraic structure of Weihrauch degrees. Logical Methods in Computer Science, to appear.

[10] Vasco Brattka and Tahina Rakotoniaina. On the uniform computational content of Ramsey's theorem. *J. Symb. Log.*, 82(4):1278–1316, 2017.

[11] Keh Hsun Chen. Recursive well-founded orderings. *Ann. Math. Logic*, 13(2):117–147, 1978.

[12] Peter A. Cholak, Carl G. Jockusch, and Theodore A. Slaman. On the strength of Ramsey's theorem for pairs. *J. Symbolic Logic*, 66(1):1–55, 2001.

[13] Chris J. Conidis. Comparing theorems of hyperarithmetic analysis with the arithmetic Bolzano-Weierstrass theorem. *Trans. Amer. Math. Soc.*, 364(9):4465–4494, 2012.

[14] Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Fifth edition.

[15] François G. Dorais, Damir D. Dzhafarov, Jeffry L. Hirst, Joseph R. Mileti, and Paul Shafer. On uniform relationships between combinatorial problems. *Trans. Amer. Math. Soc.*, 368(2):1321–1359, 2016.

[16] Damir D. Dzhafarov. Cohesive avoidance and strong reductions. *Proc. Amer. Math. Soc.*, 143(2):869–876, 2015.

[17] Damir D. Dzhafarov, Jun Le Goh, Denis R. Hirschfeldt, Ludovic Patey, and Arno Pauly. Ramsey's theorem and products in the Weihrauch degrees. arXiv:1804.10968, 2018.

[18] Harvey Friedman. Some systems of second order arithmetic and their use. *Proceedings of the International Congress of Mathematicians (Vancouver, B. C., 1974), Vol. 1*, pages 235–242, 1975.

[19] Harvey M. Friedman and Jeffry L. Hirst. Weak comparability of well orderings and reverse mathematics. *Ann. Pure Appl. Logic*, 47(1):11–29, 1990.

[20] Noam Greenberg and Antonio Montalbán. Ranked structures and arithmetic transfinite recursion. *Trans. Amer. Math. Soc.*, 360(3):1265–1307, 2008.

[21] R. Halin. Über die Maximalzahl fremder unendlicher Wege in Graphen. *Math. Nachr.*, 30:63–85, 1965.

[22] Joseph Harrison. Recursive pseudo-well-orderings. *Trans. Amer. Math. Soc.*, 131:526–543, 1968.

[23] Kojiro Higuchi and Arno Pauly. The degree structure of Weihrauch reducibility. *Log. Methods Comput. Sci.*, 9(2):2:02, 17, 2013.

[24] Denis R. Hirschfeldt and Carl G. Jockusch, Jr. On notions of computability-theoretic reduction between Π^1_2 principles. *J. Math. Log.*, 16(1):1650002, 59, 2016.

[25] Jeffry Hirst and Carl Mummert. Using Ramsey's theorem once. arXiv:1611.03134, 2 Jun 2017.

[26] Jeffry Lynn Hirst. *Combinatorics in Subsystems of Second Order Arithmetic.*

ProQuest LLC, Ann Arbor, MI, 1987. Thesis (Ph.D.)–The Pennsylvania State University.

[27] C. G. Jockusch, Jr. and T. G. McLaughlin. Countable retracing functions and Π^0_2 predicates. *Pacific J. Math.*, 30:67–93, 1969.

[28] Takayuki Kihara, Alberto Marcone, and Arno Pauly. Searching for an analogue of ATR in the Weihrauch lattice. arXiv:1812.01549, 2018.

[29] Richard Laver. On Fraïssé's order type conjecture. *Ann. of Math. (2)*, 93:89–111, 1971.

[30] Antonio Montalbán. Indecomposable linear orderings and hyperarithmetic analysis. *J. Math. Log.*, 6(1):89–120, 2006.

[31] Antonio Montalbán. On the Π^1_1-separation principle. *MLQ Math. Log. Q.*, 54(6):563–578, 2008.

[32] Itay Neeman. The strength of Jullien's indecomposability theorem. *J. Math. Log.*, 8(1):93–119, 2008.

[33] Itay Neeman. Necessary use of Σ^1_1 induction in a reversal. *J. Symbolic Logic*, 76(2):561–574, 2011.

[34] Ludovic Patey. The weakness of being cohesive, thin or free in reverse mathematics. *Israel J. Math.*, 216(2):905–955, 2016.

[35] Arno Pauly. On the (semi)lattices induced by continuous reducibilities. *MLQ Math. Log. Q.*, 56(5):488–502, 2010.

[36] Arno Pauly. Computability on the countable ordinals and the Hausdorff-Kuratowski theorem. *CoRR*, abs/1501.00386, 2015.

[37] Klaus-Peter Podewski and Karsten Steffens. Injective choice functions for countable families. *J. Combinatorial Theory Ser. B*, 21(1):40–46, 1976.

[38] Hartley Rogers, Jr. *Theory of recursive functions and effective computability*. MIT Press, Cambridge, MA, second edition, 1987.

[39] Gerald E. Sacks. *Higher recursion theory*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1990.

[40] Richard A. Shore. On the strength of Fraïssé's conjecture. In *Logical methods (Ithaca, NY, 1992)*, volume 12 of *Progr. Comput. Sci. Appl. Logic*, pages 782–813. Birkhäuser Boston, Boston, MA, 1993.

[41] Stephen G. Simpson. On the strength of König's duality theorem for countable bipartite graphs. *J. Symbolic Logic*, 59(1):113–123, 1994.

[42] Stephen G. Simpson. *Subsystems of second order arithmetic*. Perspectives in Logic. Cambridge University Press, Cambridge; Association for Symbolic Logic, Poughkeepsie, NY, second edition, 2009.

[43] John R. Steel. Forcing with tagged trees. *Ann. Math. Logic*, 15(1):55–74, 1978.

[44] Robert Alan Van Wesep. *Subsystems of Second-order Arithmetic, and Descriptive Set Theory under the Axiom of Determinateness*. ProQuest LLC, Ann Arbor, MI, 1977. Thesis (Ph.D.)–University of California, Berkeley.