Applications of Computability Theory to Infinitary Combinatorics

Noah A. Hughes, Ph.D.

University of Connecticut, 2021

ABSTRACT

Reverse mathematics is a rich framework for benchmarking the relative proof theoretic strength of classical mathematics (that fragment of mathematics which is sufficiently realized in a countable setting.) This thesis continues the reverse mathematics program by classifying several previously unstudied theorems of combinatorics in terms of the *big five* subsystems.

We also analyze theorems from the reverse mathematics zoo using the relatively novel computability-theoretic reducibilities associated with strong reductions and Weihrauch complexity. In the course of this work we will establish several examples of reverse mathematically equivalent principles which are distinguished under some computability-theoretic reduction.

We take principles from infinitary combinatorics as our object of study. Chapters 2 and 4 investigate theorems regarding infinite graphs and hypergraphs, while chapter 3 investigates theorems about infinite partially ordered sets.

Applications of Computability Theory to Infinitary Combinatorics

Noah A. Hughes

M.A., Appalachian State University, North Carolina, May 2015

B.S., Appalachian State University, North Carolina, May 2014

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2021

APPROVAL PAGE

Doctor of Philosophy Dissertation

Applications of Computability Theory to Infinitary Combinatorics

Presented by

Noah A. Hughes, B.S., M.A.

Major Advisor ______________________
Damir D. Dzhafarov

Associate Advisor ______________________
David Reed Solomon

Associate Advisor ______________________
Thomas W. Roby

University of Connecticut

2021

ACKNOWLEDGMENTS

I first wish to thank my advisor Damir Dzhafarov. Not only has he taught me a great deal of mathematics, he has also provided valuable insight into the practice of a working mathematician. I also wish to thank my associate advisors Reed Solomon and Tom Roby. I thank Reed for the wealth of knowledge he has shared with me over the years, and for the clarity and patience with which he has addressed my many questions. I thank Tom for his careful reading of my dissertation, for the lessons he has taught me in communicating mathematics, and for his inspiring approach to pedagogy. I owe a great deal to the entire University of Connecticut Department of Mathematics, but I wish to specifically thank Monique Roy for the compassion and care she brings to the department. I am also very grateful to Jeff Hirst for originally piquing my interest in mathematical logic, and suggesting that I attend UConn. And to each of my peers who worked alongside me in Monteith 322: your advice, support, and comradery have proven invaluable.

I especially wish to thank my wife, Madelyn Hughes, for her constant faith in me. Finally, I'd like to say thank you on behalf of my work and myself, and I hope I've passed the audition.

Contents

Chapter 1

Introduction: analyzing mathematical principles in mathematical logic

This thesis is a contribution to the central practice in mathematical logic of analyzing classical mathematical principles in terms of their proof-, set-, and computability-theoretic complexity in order to discern underlying connections between disparate areas of mathematics as well as the nature of mathematical proof. The principles studied herein arise in infinitary combinatorics.

We utilize two robust frameworks throughout to conduct our analysis. The first is *reverse mathematics,* a proof-theoretic program in which classical theorems of mathematics are bench-marked against subsystems of second-order arithmetic linearly ordered by set-comprehension. The second framework, which we refer to as *Weihrauch analysis*, is a collection of computability-theoretic reductions between mathematical principles formulated as formal *problems*.

In chapter 2, we use reverse mathematics to benchmark several principles related to matchings on a bipartite graph. In chapter 3, we distinguish reverse-mathematically equivalent principles regarding infinite partial orders via the computability-theoretic reductions from Weihrauch analysis. In chapter 4, we report on preliminary results regarding the reverse-mathematical and Weihrauch complexity of principles discussing unique colorability in hypergraphs.

As the techniques used in this work can all be motivated from a computability-theoretic perspective, we begin with an introduction to the fundamentals of computability theory before discussing the specifics of our chosen frameworks.

1.1 Computability Theory

Computability theory arose in the early 20th century to give a rigorous mathematical formulation of the notion of *algorithm* or computational procedure. There are many equivalent ways to define a formal computer which carries out a prescribed algorithm but it is widely agreed that Turing's formulation of what are now known as *Turing machines* [25] yields the essential definition of an algorithm or computational procedure. We recommend Weber [26] for a friendly introduction to the basic notions of computability theory and Soare [23] for an in-depth discussion.

We call a function $f : \mathbb{N} \to \mathbb{N}$ *computable* if there is a Turing machine which given input n, outputs $f(n)$. If $f(n)$ is defined for all $n \in \mathbb{N}$, we say f is *total*, else we call f *partial*. As is standard, we fix an effective enumeration of all partial computable functions $\Phi_0, \Phi_1, \dots \Phi_e, \dots$ and assume that given the triple (e, n, s), we can run Turing machine e on input n for s steps via a universal Turing machine. We abbreviate this computation with the symbol $\Phi_e(n)[s]$. If $\Phi_e(n)[s]$ is defined, we write $\Phi_e(n)[s]\downarrow$ and say Φ_e *halts* on input n in s steps. Otherwise we write $\Phi_e(n)[s]\uparrow$. We call the least such s for which $\Phi_e(n)[s]\downarrow$ the *use* of $\Phi_e(n)$ and denote it by $\varphi_e(n) = s$. If no such s exists, we write $\Phi_e(n)\uparrow$ and say Φ_e *diverges* on input n. The statement that such an s exists is abbreviated by $\Phi_e(n)\downarrow$.

We call a set $A \subseteq \mathbb{N}$ *computable* if its characteristic function χ_A is computable. That is, there is some e such that $\Phi_e = \chi_A$. Note $\chi_A \in 2^{\mathbb{N}}$. We freely conflate a set, its characteristic function, and the binary sequence it defines.

We assume without loss of generality that each of $\Phi_0, \Phi_1, \ldots$ are computed by *oracle Turing machines* with the empty set $\emptyset$ used as an oracle. Then for any set X, we have the partial computable functions *relative to* X, $\Phi_0^X, \Phi_1^X, \ldots$ and define $\Phi_e^X(n)[s]\downarrow$, $\varphi_e^X(n) = s$ and $\Phi_e^X(n)[s]\uparrow$ similarly. We say A is *computable in*, or *computable relative to*, X if there is an index e such that $\Phi_e^X = \chi_A$. We write $A \leq_T X$ in this case and say A is *Turing reducible* to X. If in addition $X \leq_T A$, we write $A \equiv_T X$ and say A is *Turing equivalent* to X. It is easy to see that $\equiv_T$ is an equivalence relation and that $\leq_T$ forms an upper semi-lattice on the equivalence classes of $\equiv_T$. The least upper-bound under $\leq_T$ of A and X is the *join* $A \oplus X = \{2n : n \in A\} \cup \{2n+1 : n \in X\}$.

If Φ_e is a total function and the range of Φ is contained in $\{0, 1\}$, it is natural to regard Φ_e as a *functional* mapping 'sets' X in 2^ω to other sets Y via $\chi_Y = \Phi_e^X$.

The canonical example of set which is not computable is the *halting set*

$$\emptyset' = \{e : \exists s\, \Phi_e(e)[s]\downarrow\}.$$

To show $\emptyset'$ is not computable, we use a *diagonalization argument*: we show that no partial computable function Φ_e can define the set in question. These arguments will be crucial in Chapter 3.

Example 1.1.1. The set $\emptyset'$ is not computable.

Proof. Assume by way of contradiction that there is an index e such that $\Phi_e = \chi_{\emptyset'}$. Define a partial computable function Ψ by

$$\Psi(n) = \begin{cases} 0 & \text{if } \Phi_e(n) = 0 \\ \uparrow & \text{if } \Phi_e(n) = 1. \end{cases}$$

Note Ψ is partial computable and thus there is an index e_0 such that $\Phi_{e_0} = \Psi$. Then $\Phi_{e_0}(e_0)\downarrow$ if and only if $e_0 \notin \emptyset'$, a contradiction. $\square$

The use of the term *diagonalization* to describe this argument arises from the definition of $\emptyset'$. In a general construction, we say we have *diagonalized* Φ_e, if we have avoided it defining the object in question.

We briefly mention that we may effectively code finite strings $\sigma \in \mathbb{N}^{<\mathbb{N}}$ by natural numbers. For instance, we may code σ by the number $2^{\sigma(0)+1}3^{\sigma(1)+1}\cdots p_n^{\sigma(n)+1}$ where p_n is the nth prime and σ is a string of length $n+1$. Similarly, we may code finite sets as natural numbers and infinite objects, e.g., n-ary functions, as sets of natural numbers. In what follows, the specifics of coding will not matter outside of it being effective. For details on coding in computability theory see Soare [23] and for details on coding in Reverse Mathematics, see Simpson [22]. We now freely consider n-ary computable functions via codes for finite sequences.

We say a relation $R(x_0, \ldots, x_n)$ on $\mathbb{N}^{n+1}$ is *computable* if there is a total computable function Φ_e with range in $\{0,1\}$ such that $\Phi_e(x_0,\ldots,x_n)\downarrow= 1$ if and only if $R(x_0,\ldots,x_n)$ holds. We say R is X-*computable* if the same is true of a computable function Φ_e^X.

Definition 1.1.2 (The arithmetical hierarchy). 1. We say a set A is Σ_0^0 (Π_0^0) if it is computable.

2. For $n \geq 1$, we say A is Σ_n^0 if there is a computable predicate $R(y, x_0, \ldots, x_n)$ such that

$$y \in A \iff \exists x_0 \forall x_1 \cdots Q x_n R(y, x_0, \ldots, x_n)$$

where Q is $\exists$ if n is even and $\forall$ if n is odd. Similarly, we say A is Π_n^0 if there is a computable predicate $R(y, x_0, \ldots, x_n)$ such that

$$y \in A \iff \forall x_0 \exists x_1 \cdots Q x_n R(y, x_0, \ldots, x_n)$$

where Q is $\forall$ if n is even and $\exists$ if n is odd.

3. We say A is Δ_n^0 if A is both Σ_n^0 and Π_n^0.

4. We say A is Σ_n^0 (Π_n^0) *in* X if R is an X-computable predicate.

1.2 Reverse Mathematics

The basic idea of reverse mathematics is as follows: given a mathematical theorem P, we seek to find axioms which are both necessary and sufficient to prove P. The way we do this is by first establishing that P is *not* provable in some weak base theory $\mathcal{B}$. We then find a set-existence axiom $\mathcal{A}$ which when added to the weak base theory, suffices to prove P. So we show

$$\mathcal{B} + \mathcal{A} \vdash \mathsf{P},$$

which yields that $\mathcal{A}$ is *sufficient* to prove P, over $\mathcal{B}$. To show that $\mathcal{A}$ is *necessary*, we 'reverse' the mathematical practice of proving theorems from axioms by deriving the axiom $\mathcal{A}$ from the theorem P. That is, we show

$$\mathcal{B} + \mathsf{P} \vdash \mathcal{A}.$$

This step is called a *reversal* and it yields that $\mathcal{A}$ is in fact *necessary* to prove P over $\mathcal{B}$.

The remarkable phenomenon within reverse mathematics is that, with only four distinct set existence axioms, we can carry this out analysis for an immense amount of classical mathematics over one fixed base theory. We review these *big five* axiom systems and suggest Simpson [22] for an in-depth discussion of each.

The majority of reverse mathematics takes place within *second-order arithmetic*, denoted Z_2, which is a two-sorted formal system capable of formalizing a large portion of classical mathematics via countable codes. As mentioned above, the encodings we use will be unimportant save that we can effectively manipulate codes. To see a detailed discussion of how objects are encoded in second-order arithmetic we recommend Simpson [22] and Hirst [16].

The language L_2 of second-order arithmetic is two-sorted with *number variables* $x, y, z, \ldots$ intended to range over ω, and *set variables* $X, Y, Z, \ldots$ intended to range over $\mathcal{P}(\omega)$, the power-set of ω. We have the usual symbols $+$, $\cdot$ and $<$ of first-order arithmetic as well as the symbol $\in$ for membership and constant symbols 0 and 1. We build *formulas* in L_2 by recursion on the atomic terms, which include all number and set variables as well as $t_1 = t_2$, $t_1 + t_2$, $t_1 \cdot t_2$, $t_1 < t_2$ and $x \in X$. A *sentence* is a formula with no free variables.

An L_2 *structure* is of the form $(N, S, +_N, \cdot_N, <_N, 0_N, 1_N)$ where $(N, +_N, \cdot_N, <_N, 0_N, 1_N)$ is a structure in the language of first-order arithmetic and $S \subseteq \mathcal{P}(N)$, the power-set of N. We call N the first-order part of the structure and S the second-order part. In any structure, $\in$ is interpreted as true membership. We are now ready to define the axiom system Z_2.

Definition 1.2.1 (Second-order arithmetic)**.** The *axioms of second-order arithmetic* consist of

1. the collection PA^- of axioms of a discrete ordered commutative semiring;

2. the induction scheme, consisting of all instances of

$$\big[\theta(0) \wedge \forall n(\theta(n) \to \theta(n+1))\big] \to \forall n\, \theta(n))$$

   where $\theta(n)$ is any formula of L_2; and

3. the comprehension scheme, consisting of all instances of

$$\exists X \forall n(n \in X \leftrightarrow \theta(n))$$

   where $\theta(n)$ is any formula of L_2 in which X does not occur freely.

The intended model for Z_2 is, of course,

$$(\omega, \mathcal{P}(\omega), +, \cdot, <, \in).$$

Any model with first-order part ω is called an ω-model. There are nonstandard models of Z_2 and its subsystems, in which the first-order part $N \neq \omega$. To maintain generality, we reserve the symbol $\mathbb{N}$ to denote the set $\{x : x = x\}$ in Z_2 or any subsystem and note that $\mathbb{N}$ may not be the standard natural numbers ω.

We obtain *subsystems of second-order arithmetic* by restricting the induction and comprehension scheme to specific classes of formulas.

Definition 1.2.2.

1. For any L_2 formula ψ, we abbreviate $\exists x(x < y \wedge \psi)$ and $\forall x(x < y \to \psi)$ as $(\exists x < y)(\psi)$ and $(\forall x < y)(\psi)$ respectively, and call these quantifiers *bounded.*

2. We say θ is a Σ^0_0 (Π^0_0) *formula* if it contains only bounded quantifiers.

3. For $n \geq 1$, we say θ is a Σ^0_n *formula* if it is equivalent to one of the form $\exists x_0 \forall x_1 \cdots Q x_n \psi$, where ψ is a Σ^0_0 formula, and Q is $\exists$ if n is even and $\forall$ if n is odd.

4. For $n \geq 1$, we say θ is a Π^0_n *formula* if it is equivalent to one of the form $\forall x_0 \exists x_1 \cdots Q x_n \psi$, where ψ is a Σ^0_0 formula,and Q is $\forall$ if n is even and $\exists$ if n is odd.

5. For $n \geq 0$, we say θ is a Δ^0_n *formula* if it is both Σ^0_n and Π^0_n.

6. We say θ is *arithmetical* if it has no set quantifiers.

7. For $n \geq 1$, we say θ is a Σ^1_n *formula* if it is equivalent to one of the form $\exists X_0 \forall X_1 \cdots Q X_n \psi$, where ψ is an arithmetical formula, and Q is $\exists$ if n is even and $\forall$ if n is odd.

8. For $n \geq 1$, we say θ is a Π^1_n *formula* if it is equivalent to one of the form $\forall X_0 \exists X_1 \cdots Q X_n \psi$, where ψ is an arithmetical formula, and Q is $\forall$ if n is even and $\exists$ if n is odd.

9. For $n \geq 0$, we say θ is a Δ^1_n *formula* if it is both Σ^1_n and Π^1_n.

The similarities between these classes of formulas and the arithmetical hierarchy is no coincidence. A subset $A \subseteq \omega$ is Σ^0_n (Π^0_n) if it is *definable* by a Σ^0_n (Π^0_n) formula with no set variables, meaning there is some Σ^0_n (Π^0_n) formula $\theta(n)$ with no set variables such that $A = \{n \in \omega : \theta(n)\}$. The analogous hierarchy of sets definable by Σ^1_n and Π^1_n is the *analytic hierarchy*, but we will not need it.

The base theory RCA_0 is obtained by restricting the induction scheme to Σ^0_1 formulas and the comprehension scheme to Δ^0_1 formulas.

Definition 1.2.3. The axiom system RCA_0 consists of the collection PA^- with the induction scheme of Z_2 restricted to Σ^0_1 formulas and the comprehension scheme restricted to Δ^0_1 formulas.

We refer to the comprehension scheme of RCA_0 as *recursive comprehension.* The ω-models of RCA_0 are exactly those in which the second-order part is a *Turing ideal*, i.e., closed under join and Turing reducibility. Thus we consider RCA_0 the fragment of Z_2 loosely corresponding to "computable mathematics."

We next define the other four of the big five subsystems, all of which are obtained by adding axioms to RCA_0. A *tree* is any set of finite strings closed under prefix. We will work exclusively with subtrees of $2^{<\mathbb{N}}$ or $\mathbb{N}^{<\mathbb{N}}$, the set of finite strings from $\{0, 1\}$ or $\mathbb{N}$ respectively. We use λ to denote the empty string.

Definition 1.2.4. The axiom system WKL_0 consists of the axioms of RCA_0 together with the statement of *Weak Kőnig's Lemma*:

$$\text{Every infinite subtree of } 2^{<\mathbb{N}} \text{ has an infinite path.}$$

The statement of weak Kőnig's lemma can be seen as the assertion that the Cantor space 2^ω is compact. Thus WKL_0 is the weakest axiom system in which we may carry out compactness arguments.

Definition 1.2.5. The axiom system ACA_0 consists of the axioms of RCA_0 with the comprehension scheme restricted to arithmetical formulas.

We refer to the comprehension scheme of ACA_0 as *arithmetic comprehension*. Arithmetic comprehension can be recast in computability theoretic terms as the statement that for every set X, the *Turing jump* $X' = \{e : \Phi_e^X(x)\downarrow\}$ exists.

To define the next system, ATR_0, we require a few preliminary formulas. Let $\mathrm{WO}(X)$ and $\mathrm{LO}(X)$ be L_2-formulas which respectively say X is a countable well-order and X is a countable linear order. If X is any reflexive relation, let $\mathrm{field}(X)$ denote the domain of the relation, that is, $\mathrm{field}(X) = \{i : (i,i) \in X\}$.

Definition 1.2.6. The formula $\mathrm{H}_\theta(X,Y)$ says that $\mathrm{LO}(X)$ and

$$Y = \{(n,j) : j \in \mathrm{field}(X) \wedge \theta(n, Y^j)\}$$

where

$$Y^j = \{(m,i) : i <_X j \wedge (m,i) \in Y\}.$$

Intuitively, H_θ says that Y is the set resulting from iterating the arithmetical formula θ along the well-ordering X.

Definition 1.2.7. The axiom system ATR_0 consists of ACA_0 together with all instances of

$$\forall X \left(\mathrm{WO}(X) \to \exists Y\, \mathrm{H}_\theta(X,Y)\right)$$

where θ is an arithmetical formula.

The axioms of ATR_0 allow transfinite constructions to be carried out using repeated applications of arithmetic comprehension. This is the weakest system in which a theory of ordinals can be sufficiently developed.

The last of the big five subsystems is Π^1_1-CA_0 and it appends comprehension for all Π^1_1 definable sets.

Definition 1.2.8. The axiom system Π^1_1-CA_0 consists of ACA_0 together with the comprehension scheme restricted to Π^1_1 formulas.

We refer to the comprehension scheme of Π^1_1-CA_0 as Π^1_1 *comprehension*.

Each of these systems is strictly stronger than the previous, yielding the following picture of the big five. See Simpson [22] for proofs of this fact.

$$\mathsf{RCA}_0 \Leftarrow \mathsf{WKL}_0 \Leftarrow \mathsf{ACA}_0 \Leftarrow \mathsf{ATR}_0 \Leftarrow \Pi^1_1\text{-}\mathsf{CA}_0.$$

1.3 Problems and Reductions

When discussing computability theoretic reductions, we restrict our attention implicitly to ω-models. That is, we work with ω and its subsets and not the more general set $\mathbb{N}$. The combinatorial principles we study all have a Π^1_2-gestalt in the following way. Consider the sentence

$$\forall X(\varphi(X) \to \exists Y \psi(X, Y))$$

where φ and ψ are arithmetical formulas. We think of this as a mathematical principle P which asserts that for every object X satisfying φ, there is another object Y which relates to X according to ψ. These sorts of principles abound in mathematics: every non-zero ring with identity contains a maximal ideal; every infinite covering of $[0,1]$ with a sequence of open intervals contains a finite subcovering; every infinite subtree of $2^{<\mathbb{N}}$ contains an infinite path. It is fruitful to call such a principle a *problem* which consists of *instance-solution pairs* (X,Y). Here X is an *instance of* P if and only if $\varphi(X)$ and Y is a *solution* to the instance X if and only if $\psi(X,Y)$. In this way, we can see any ring R with identity as an instance of the problem "every non-zero ring with identity contains a maximal ideal". Then any maximal ideal $M \subseteq R$ is a solution to R. Of course, any given instance may have many solutions.

This formulation can be extended to a very general setting in which any multifunction between represented spaces is considered a problem. This is the usual approach taken in computable analysis, but as all of our principles will naturally be formulated with ω and its subsets, we do not require this generality. We recommend Brattka, Gherardi and Pauly [2] for an introduction to this formulation and an overview of its applications in computable analysis.

Given two problems P and Q we establish *reductions* between them as follows: we say P is *reducible* to Q if given any instance X_P of P, there is a way to transform it (perhaps computably) into an instance X_Q of Q, such that any solution Y_Q to X_Q can then be transformed into a solution Y_P of the original instance X_P of P. There are four specific notions of reducibility that we primarily utilize in our analysis.

Definition 1.3.1. Given two problems P and Q, we say

1. P is *computably reducible* to Q, written $\mathsf{P} \leq_\mathsf{c} \mathsf{Q}$, if and only if given any instance X_P of P, there is an instance X_Q of Q such that $X_\mathsf{Q} \leq_T X_\mathsf{P}$, and for any solution Y_Q of X_Q, there is a solution Y_P of X_P such that $Y_\mathsf{P} \leq_T X_\mathsf{P} \oplus Y_\mathsf{Q}$.

2. P is *strongly computably reducible* to Q, written $\mathsf{P} \leq_{\mathsf{sc}} \mathsf{Q}$, if and only if given any instance X_{P} of P, there is an instance X_{Q} of Q such that $X_{\mathsf{Q}} \leq_T X_{\mathsf{P}}$, and for any solution Y_{Q} of X_{Q}, there is a solution Y_{P} of X_{P} such that $Y_{\mathsf{P}} \leq_T Y_{\mathsf{Q}}$.

3. P is *Weihrauch reducible* to Q, written $\mathsf{P} \leq_{\mathsf{W}} \mathsf{Q}$, if and only if there are two fixed Turing functionals Φ and Ψ such that, given any instance X_{P} of P, the set defined by $\Phi^{X_{\mathsf{P}}}$ is an instance of Q, and any solution Y_{Q} of this instance has that $\Psi^{X_{\mathsf{P}} \oplus Y_{\mathsf{Q}}}$ defines a solution to X_{P}.

4. P is *strongly Weihrauch reducible* to Q, written $\mathsf{P} \leq_{\mathsf{sW}} \mathsf{Q}$, if and only if there are two fixed Turing functionals Φ and Ψ such that, given any instance X_{P} of P, the set defined by $\Phi^{X_{\mathsf{P}}}$ is an instance of Q, and any solution Y_{Q} of this instance has that $\Psi^{Y_{\mathsf{Q}}}$ defines a solution to X_{P}.

Weihrauch reducibility is sometimes referred to as *uniform reducibility*. This is because the reduction procedures Φ and Ψ are fixed. In a computable reduction, we may utilize separate Turing reductions for each instance of P or each solution of the computed instance of Q. Thus showing $\mathsf{P} \not\leq_{\mathsf{W}} \mathsf{Q}$ formally verifies that any proof of P using Q will require non-uniform decisions to be made.

The omission of the join in strong reductions is non-trivial. Consider the simple problems $\mathsf{P} : \forall X \exists Y (Y = X)$ and $\mathsf{Q} : \forall X \exists Y (Y = \emptyset)$. It is reasonable to think the first problem should be reducible to any problem as we simply need record what X is to yield a solution. But notice P is not strongly reducible to the second problem Q because any non-computable set, say $\emptyset'$, is an instance of P. No matter what instance of Q we compute from $\emptyset'$, the resulting solution will be $\emptyset$ and thus will be unable to compute the solution, $\emptyset'$, of this instance of P. If the join is permitted, then we see P is reducible to Q, as $X \leq_T X \oplus Y$ for any set Y.

We summarize below in the leftmost diagram the implications between these reductions. While these are easy to show, they are in fact strict. We will see examples of problems P and Q that witness this in Chapter 3. In the rightmost diagram, we give a graphical representation of a strong Weihrauch reduction $\mathsf{P} \leq_{\mathsf{sW}} \mathsf{Q}$, where Φ and Ψ are fixed Turing functionals, and the dotted lines indicate an instance-solution pair.

$$\begin{tikzcd} & \leq_{\mathsf{sW}} \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\ \leq_{\mathsf{W}} \arrow[dr, Rightarrow] & & \leq_{\mathsf{sc}} \arrow[dl, Rightarrow] \\ & \leq_{\mathsf{c}} & \end{tikzcd} \qquad \begin{tikzcd} \mathsf{P} \arrow[r, phantom, "\leq_{\mathsf{sW}}"] & \mathsf{Q} \\ X_{\mathsf{P}} \arrow[r, "\Phi"] \arrow[d, dotted] & X_{\mathsf{Q}} \arrow[d, dotted] \\ Y_{\mathsf{P}} & Y_{\mathsf{P}} \arrow[l, "\Psi"'] \end{tikzcd}$$

For Theorem 3.1.6 below, we will make use of a *generalized Weihrauch reduction*. We include the definition here for completeness. Intuitively, P is Weihrauch reducible to Q in the generalized sense, written $P \leq_{\mathsf{gW}} Q$, if any instance of P can be uniformly solved with multiple uses of Q. There are several equivalent ways to formally define this notion but we elect to use the game theoretic approach of Hirschfeldt and Jockusch [12] (see Definitions 4.1 and 4.3).

We define the *n-fold join* of sets $X_0, \dots, X_n$ by

$$\bigoplus_{i \leq n} X_i = \{n\} \oplus \{(i,k) : i \leq n \wedge k \in X_i\}.$$

Note that we may effectively recover n from $\bigoplus_{i \leq n} X_i$. When it is clear from context, we will write X_0 for $\bigoplus_{i \leq 0} X_i$ and $X_0 \oplus X_1$ for $\bigoplus_{i \leq 1} X_i$.

Definition 1.3.2. Given problems P and Q, the *reduction game* $G(\mathsf{Q} \to \mathsf{P})$ is a two-player game ending when one player wins. The game is played as follows.

For the first move, Player I begins by playing a P-instance X_0. Player II responds by either playing an X_0-computable solution to X_0, in which case they win, or by playing a Q-instance $Y_n \leq_T X_0$. If Player II cannot respond, Player I wins.

On the nth move, for $n > 1$, Player I plays a solution X_{n-1} to the Q-instance Y_{n-1}. Player II responds by either playing a $(\bigoplus_{i<n} X_i)$-computable solution to X_0, in which case they win, or by playing a Q-instance $Y_n \leq_T (\bigoplus_{i<n} X_i)$.

If there is no move at which Player II wins, then Player I wins.

Definition 1.3.3. A *computable strategy* for Player II in the reduction game $G(\mathsf{Q} \to \mathsf{P})$ is a Turing functional which, when given the join of the first n moves of Player I, outputs the nth move for Player II. Specifically, the strategy is a functional Φ such that, if Z is the join of the first n moves of Player I, then $\Phi^Z = V \oplus Y$ where Y is the nth move of Player II, and $V = \{1\}$ if Player II wins with Y, or $V = \emptyset$ otherwise. The strategy is *winning* if it allows Player II to win no matter what moves Player II makes.

Definition 1.3.4. We say P is *Weihrauch reducible to* Q *in the generalized sense*, $\mathsf{P} \leq_{\mathsf{gW}} \mathsf{Q}$, if there is a winning computable strategy for Player II in the game $G(\mathsf{Q} \to \mathsf{P})$.

The figure below illustrates how the notion $\mathsf{P} \leq_{\mathsf{gW}} \mathsf{Q}$ relates to those given in Definition 1.3.1. In general, as Theorem 3.1.6 witnesses, $\mathsf{P} \leq_{\mathsf{gW}} \mathsf{Q}$ does not imply $\mathsf{P} \leq_{\mathsf{c}} \mathsf{Q}$.

$$
\begin{tikzcd}
 & \leq_{\mathsf{sW}} \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\
\leq_{\mathsf{W}} \arrow[d, Rightarrow] \arrow[dr, Rightarrow] & & \leq_{\mathsf{sc}} \arrow[dl, Rightarrow] \\
\leq_{\mathsf{gW}} & \leq_{\mathsf{c}} &
\end{tikzcd}
$$

Chapter 2

Matching Problems

In this chapter, we investigate a generalization of Hall's theorem for matchings on bipartite graphs. We abstract the notion of a bipartite graph to a *matching problem* $P = (A, B, R)$ where A and B are subsets of $\mathbb{N}$, and $R \subseteq A \times B$. We call an injection $f : A \to B$ a *solution* of P if $(a, f(a)) \in R$ for all $a \in A$. In other words, if A and B are disjoint then f is a matching of the bipartite graph $G = (A \cup B, R)$. Previous reverse mathematical analysis of theorems concerning the existence of solutions to matching problems have all required the assumption that for each a, the set $\{b : (a, b) \in R\}$ is finite. Equivalently, $G = (A \cup B, R)$ is a locally finite graph. In this work, we do not require this condition and find necessary and sufficient conditions for this more general matching problem to have a unique solution. We show that this result is classically a biconditional, but the relationship between each implication is more complicated in reverse mathematics. In particular, we show one direction is equivalent to ACA_0 over RCA_0 and show that the other, while provable in ACA_0, is considerably more intricate to work with. We obtain a partial reversal of this direction and study several weakenings of it.

2.1 Introduction

A *matching problem* is a triple $P = (A, B, R)$ of sets with $R \subseteq A \times B$. A *solution* to a matching problem is an injection $f : A \to B$ such that $(a, f(a)) \in R$ for all $a \in A$. If $(a, b) \in R$ we say b is a *permissible match* for a. Thus a solution f simply picks a unique permissible match for each element of A. For our purposes, we assume A and B are both subsets of $\mathbb{N}$.

To simplify notation we abbreviate the set of permissible matches of an element a by $R(a)$, that is

$$R(a) = \{b : (a, b) \in R\}.$$

For a subset $A_0 \subseteq A$, we write $R(A_0)$ for the set $\bigcup_{a \in A_0} R(a)$. If a well-order $(A, \leq_A)$ is given, we denote the initial sequence of an element a in this well order by $i_{\leq_A}(a)$, that is

$$i_{\leq_A}(a) = \{a' \in A : a' <_A a\}.$$

If the well-order is clear from context, we suppress the subscript. The sets $R(i_{\leq_A}(a))$ will be central to the work below.

Matching problems appear by several equivalent formulations in the literature. If $\mathcal{A}$ is a family of non-empty sets, a *transversal* or *system of distinct representatives* is a set containing exactly one distinct element from each $A \in \mathcal{A}$. Thus each $A \in \mathcal{A}$ is the set containing exactly its own permissible matches. This is the view from which matching problems were discussed in the early work of Philip Hall [9] and Marshall Hall [8]. Necessary and sufficient conditions were found to ensure a matching problem has at least one solution for finite A in [9] and this was extended to infinite A in [8]. A critical hypothesis in both works is that for each $a \in A$, the set $R(a)$ is finite. Equivalently no $a \in A$ may have infinitely many permissible matches. This requirement is known as *Hall's condition.*

The results of Hall [9] and Hall [8] came to be known as the *marriage theorem* with the intuition that the members of A require spouses from eligible partners in B. In these terms, the desired injection $f : A \to B$ respecting R is called an *espousing* of the society (A, B). In this view, the theorems of the two Halls were analyzed reverse mathematically by Hirst [15]. In Hirst and Hughes [19], [18] combinatorial conditions were found which exactly characterize matching problems with a unique solution and matching problems with a fixed finite number of solutions. The reverse mathematics of these conditions were also determined.

The usual framework used to study matching problems is graph theory. Indeed, every bipartite graph $G = (U \cup V, E)$ gives rise to a matching problem $P_G = (U, V, E)$ and vice versa. Hall's condition is simply the requirement that G is locally finite. For a wonderful introduction to matching from this viewpoint, we recommend Chapter 2 of Diestel [4].

In this chapter, we remove Hall's condition and seek to understand a principle about matching problems motivated from the work of Hirst and Hughes [18]. Unless specified otherwise, any matching problem $P = (A, B, R)$ below may contain $a \in A$ for which $R(a)$ is an infinite set.

Note a *total-order* $(A, \leq_A)$ is a set A with a homogeneous relation $\leq_A$ such that $\leq_A$ is transitive, antisymmetric, and for every pair $a, a' \in A$, either $a \leq_A a'$ or $a' \leq_A a$. We call $(A, \leq_A)$ a *well-order* if every non-empty subset of A contains a least element with respect to $\leq_A$.

Theorem 2.1.1. *Let $P = (A, B, R)$ be a matching problem. There is a well-order $(A, \leq_A)$ such that for each $a \in A$, there is a unique $b \in B$ satisfying*

$$R(a) - R(i_{\leq_A}(a)) = \{b\},$$

if and only if P has a unique solution.

Proof. Fix $P = (A, B, R)$. For the only if direction, suppose we have a well-order $(A, \leq_A)$ as hypothesized and define $f : A \to B$ by $f(a) = b$, where b is the unique element in $R(a) - R(i(a))$. Then f is the unique solution of P. To see that f is indeed a solution, note if f were not injective witnessed by say $a' \leq_A a$, then $f(a') \in R(i(a))$ and $f(a') \in R(a) - R(i(a))$, a contradiction. To see that it is unique, suppose $g : A \to B$ is a distinct solution. Fix the $\leq_A$-least a such that $g(a) \neq f(a)$. So $g(a) \in R(i(a))$. Let $D = \{a' <_A a : g(a) \in R(a)\}$. Since g is injective, $g(a) \neq f(a') = g(a')$ for any $a' \in D$. Now take a', the $\leq_A$-least element of D. So $g(a) \notin R(i(a'))$ and $g(a) \neq f(a')$. This implies that $R(a') - R(i(a')) = \{f(a'), g(a)\}$, a contradiction.

For the if direction, let f be the unique solution of P. Define a binary relation $\sqsubseteq$ on A by $a' \sqsubseteq a$ if and only if $f(a') \in R(a)$. Since f is a unique solution of P, it follows that $\sqsubseteq$ is reflexive, anti-symmetric and acyclic. Let $\leq'_A$ be the transitive closure of $\sqsubseteq$. Then $\leq'_A$ is a partial order. We claim every chain in $\leq'_A$ is well-founded. Suppose not: then there is an infinite descending chain $A_0 = \{a_0 \geq'_A a_1 \geq'_A \cdots\}$. By expanding A_0 with finite chains in $\sqsubseteq$ if necessary, assume for each n that $a_n \sqsupseteq a_{n+1}$. Thus $f(a_{n+1}) \in R(a_n)$. Define a second solution g to P by letting $g(a) = f(a)$ if $a \notin A_0$ and $g(a_n) = f(a_{n+1})$ for each $a_n \in A_0$. This contradicts that f is unique. Since $(A, \leq'_A)$ is a well-founded partial order, we may find a well-ordering of the maximal chains of $\leq'_A$ to obtain a well-order $(A, \leq_A)$. This well-order $(A, \leq_A)$ is as desired. If not, then there is some $a \in A$ with $R(a) - R(i(a)) = \{f(a), b\}$. Note b cannot be $f(a')$ for any other $a' \in A$, since that would imply $a' \leq a$ and $b \in R(i(a))$. So we may similarly define a second solution by letting g agree with f everywhere except a, and setting $g(a) = b$. □

The naive proof of this theorem is straight-forward but requires heavy machinery. Indeed, the only if direction used induction on a general well-order while the if direction invoked the well-ordering principle. We will show below each of these proofs can be formalized in ACA_0.

For the purposes of our analysis, we divide Theorem 2.1.1 into its two implications. We use OTS and STO to label 'if' and 'only if' implications respectively.

Statement 2.1.2. STO : Let $P = (A, B, R)$ be a matching problem. If P has a unique solution, then there is a well-order $(A, \leq_A)$ such that for each $a \in A$, there is a unique $b \in B$ satisfying

$$R(a) - R(i_{\leq_A}(a)) = \{b\}.$$

Statement 2.1.3. OTS: Let $P = (A, B, R)$ be a matching problem. If there is a well-order $(A, \leq_A)$ such that for each $a \in A$, there is a unique $b \in B$ satisfying

$$R(a) - R(i(a)) = \{b\},$$

then P has a unique solution.

The acronyms abbreviate "solution then order" and "order then solution" respectively.

2.2 The principles STO and OTS

To begin our analysis, we show that OTS is provably equivalent to ACA_0 over RCA_0. The forward direction is straight-forward. For the reversal, we code the range of an arbitrary injection into the unique solution of a matching problem.

Theorem 2.2.1 (RCA_0)**.** *The following are equivalent:*

1. ACA_0

2. OTS*: Let $P = (A, B, R)$ be a matching problem. If there is a well-order $(A, \leq_A)$ such that for each $a \in A$, there is a unique $b \in B$ satisfying*

$$R(a) - R(i(a)) = \{b\},$$

then P has a unique solution.

Proof. To see that 1 implies 2, let $P = (A, B, R)$ be a matching problem with a well-order $(A, \leq_A)$ satisfying the hypothesis of 2. The set $f \subset A \times B$ where

$$(a, b) \in f \leftrightarrow [(a, b) \in R \wedge (\forall c \in A)(c <_A a \to (c, b) \notin R)]$$

is arithmetically definable with parameters A, B, R and $\leq_A$. Thus ACA_0 proves the existence of such a set.

The properties of $\leq_A$ imply that f is the desired injection from A to B. Clearly $f \subseteq R$. For each $a \in A$, there must be a $b \in B$ with $(a, b) \in f$, as $\leq_A$ guarantees a unique witness b such that $(a', b) \notin R$ for all $a' <_A a$. Since this b is unique, f is single valued. To see that f is injective, let a_1 and a_2 be two distinct elements of A.

Suppose without loss of generality that $a_1 <_A a_2$. Then by definition $f(a_2) \notin R(a_1)$, so $f(a_1) \neq f(a_2)$. Thus f is a solution to P.

It remains to show that f is a unique solution. Toward a contradiction, suppose g is a distinct solution to P. We show g cannot be injective. Let $a \in A$ be the $\leq_A$-least element such that $g(a) \neq f(a)$. If ℓ is the $\leq_A$-least element of A, then the hypothesis of 2 ensures $|R(\ell)| = 1$. So $a \neq \ell$ because both $g(a)$ and $f(a)$ are in $R(a)$. Now, $f(a)$ is the unique element of $R(a)$ that is not in $R(a')$ for any $a' <_A a$. Hence $g(a) \in R(a')$ for some $a' <_A a_0$. Fix c to be the $\leq_A$-least such element. So

$$g(a) \in R(c) - R(i(c)).$$

Then we must have $g(a) = f(c)$ by the definition of f. But as a was $\leq_A$-least such that $g(a) \neq f(a)$ and $c <_A a_0$, we have $g(a) = g(c)$, contradicting that g is injective.

To show that 2 implies 1, we let $h : \mathbb{N} \to \mathbb{N}$ be an arbitrary injection and show that the set $\operatorname{ran}(h)$ exists, which suffices by Lemma III.1.3 of Simpson [22].

To do this, we computably construct a matching problem $P = (A, B, R)$ with an appropriate ordering such that an application of 2 yields a solution which computes $\operatorname{ran}(h)$. Let $A = B = \mathbb{N}$. We construct the set R and the well-order $\leq_A$ in stages. To begin, set $R_{-1} = \leq_{-1} = F_{-1} = \emptyset$.

At stage $2s$ set

$$F_{2s} = F_{2s-1} \cup \{h(s)\};$$

$$R_{2s} = R_{2s-1} \cup \{(2s, 2s)\}; \text{ and}$$

$$\leq_{2s} = \leq_{2s-1} \cup \{(n, 2s) : n \leq 2s\}.$$

At stage $2s+1$, determine if there is an $m \leq s$ such that $m \in F_{2s}$. If so, take the least such m and set

$$\begin{aligned}
&F_{2s+1} = F_{2s} \setminus \{m\};\\
&R_{2s+1} = R_{2s} \cup \{(2m, 2s+1), (2s+1, 2m)\};\ \text{and}\\
&\leq_{2s+1} = \leq_{2s} \cup\ \{(2s+1, 2m)\} \cup \{(k, 2s+1) : k \neq 2m \wedge (k, 2m) \in\ \leq_{2s}\}\\
&\qquad\qquad \cup \{(2s+1, 2s+1)\} \cup \{(2s+1, k) : (2m, k) \in \leq_{2s}\}.
\end{aligned}$$

Note here that $R_{2s+1}(2m) = \{2m, 2s+1\}$, $R_{2s+1}(2s+1) = \{2m\}$ and

$$n_0 \leq_{2s+1} \cdots \leq_{2s+1} 2s+1 \leq_{2s+1} 2m \leq_{2s+1} \cdots \leq_{2s+1} n_{2s-1} \leq_{2s+1} 2s$$

for $n_i \leq 2s+1$.

If there is no such m, set

$$\begin{aligned}
&F_{2s+1} = F_{2s};\\
&R_{2s+1} = R_{2s} \cup \{(2s+1, 2s+1)\};\ \text{and}\\
&\leq_{2s+1} = \leq_{2s} \cup \{(k, 2s+1) : k \leq 2s+1\}.
\end{aligned}$$

Finally, let $R = \bigcup_{s\in\omega} R_s$ and $\leq_A = \bigcup_{s\in\omega} \leq_s$.

Note that both R and $\leq_A$ are computable because the membership of (m, n) or (n, m) in R or $\leq_A$ is determined by stage $\max\{m, n\}$. Since $A = B = \mathbb{N}$, we have that $P = (A, B, R)$ and $\leq_A$ exist by recursive comprehension. We claim that $(A, \leq_A)$ satisfies the hypothesis of OTS.

First, to see that $(A, \leq_A)$ is a well-order, note that each pair of successive even integers has at most two odd integers ordered between them. And each odd integer is either directly before or directly after an even integer. Hence, the maximum length of $i(k)$ for any even integer k is $2k$. Any odd integer is either directly before or after such an element, so their initial segments must also be finite. Hence $(A, \leq_A)$ has no infinite descending sequences and is thus a well-order.

Next, note for all a we have that $|R(a)| = 1$ or $|R(a)| = 2$. In the first case, either $R(a) = \{a\}$ or $R(a) = \{a'\}$ for some $a' \neq a$. If $a \in R(a)$, then $a \notin R(i(a))$ and $|R(a) - R(i(a))| = 1$ as desired. If $a' \in R(a)$, then by construction $a \leq_A a'$ and $a' \notin R(i(a))$. So again $|R(a) - R(i(a))| = 1$ as desired.

If instead $|R(a)| = 2$, we have by construction that $R(a) = \{a, a'\}$ and $R(a') = \{a\}$ with $a' \leq_A a$. Furthermore, a is not an element of $R(c)$ for all other $c \in A$. Hence, a' is the unique member of $R(a) - R(i(a))$ as needed and the claim is verified.

Apply OTS to obtain a unique solution f to $P = (A, B, R)$. Then $f(2k) = 2k$ for all $k \in \omega$, unless at some stage s, $k \in F_{2s+1}$. That is, $f(2k) = 2k$ unless k was witnessed in the range of h by some stage. Hence $n \in \operatorname{ran}(h) \iff f(2n) \neq 2n$. We see $\operatorname{ran}(h)$ is Δ^0_0-definable in f and therefore exists by recursive comprehension. This completes the proof. $\square$

The converse STO can also be proven in ACA_0 but the reversal remains elusive. The key insight towards formalizing the proof of Theorem 2.1.1 in reverse mathematics is to use the statement $\mathsf{Ext}(\omega^*)$.

Note a *poset* $(P, \leq_P)$ is a set P paired with a homogeneous relation $\leq_P$ on P which is reflexive, antisymmetric, and transitive. We call $\leq_P$ a *partial order* on P, and we say $X \subseteq P$ is a *chain* if every pair in X is comparable under $\leq_P$. We say a poset is *well-founded* if every chain contains a least element with respect to $\leq_P$.

Statement 2.2.2. $\mathsf{Ext}(\omega^*)$: If $(P, \leq_P)$ is a countable well-founded poset, then there is a well-order extending $(P, \leq_P)$, i.e., there is a well-order $(P, \leq')$ such that for all $x, y \in P$, $x \leq_P y$ implies $x \leq' y$.

The notation $\mathsf{Ext}(\omega^*)$ is based on Downey, Hirschfeldt, Lempp, and Solomon [5]. Theorem 1 of [5] states that $\mathsf{Ext}(\omega^*)$ is provable in ACA_0, implies WKL_0 over RCA_0, and is not provable in WKL_0. The exact reverse mathematical strength of $\mathsf{Ext}(\omega^*)$ is unknown. Below we show that STO implies $\mathsf{Ext}(\omega^*)$ over RCA_0. Combining this with the result of Downey, Hirschfeldt, Lempp, and Solomon yields that STO is not provable in WKL_0.

Theorem 2.2.3 (ACA_0)**.** STO*: Let $P = (A, B, R)$ be a matching problem. If P has a unique solution, then there is a well-order $(A, \leq_A)$ such that for each $a \in A$, there is a unique $b \in B$ satisfying*

$$R(a) - R(i_{\leq_A}(a)) = \{b\}.$$

Proof. We work in RCA_0 and formalize the proof of STO given in Theorem 2.1.1, noting where ACA_0 is required. Let $P = (A, B, R)$ be a matching problem with unique solution f. Define a binary relation $\sqsubseteq$ on A by $a' \sqsubseteq a$ if and only if $f(a') \in R(a)$. The relation $\sqsubseteq$ exists by recursive comprehension. Note $\sqsubseteq$ is reflexive, anti-symmetric, and acyclic because f is a unique solution. Apply ACA_0 to obtain $\leq'_A$, the transitive closure of $\sqsubseteq$. Note that $\leq'_A$ is arithmetically definable:

$$\leq'_A = \{(x, y) : \exists \langle x_1, x_2, \ldots, x_n \rangle (x = x_1 \sqsubseteq x_2 \sqsubseteq \cdots \sqsubseteq x_n = y)\}.$$

Clearly $\leq'_A$ is a partial order.

We claim $\leq'_A$ is well-founded. Suppose not: then there is an infinite descending chain $A_0 = \{a_0 \geq'_A a_1 \geq'_A \cdots\}$. Expand and relabel A_0 with finite chains from $\sqsubseteq$ if necessary to obtain that $a_n \sqsupseteq a_{n+1}$ for all n. Thus $f(a_{n+1}) \in R(a_n)$. Define a second solution g to P by letting $g(a) = f(a)$ if $a \notin A_0$, and $g(a_n) = f(a_{n+1})$ for each $a_n \in A_0$. Note that g exists by recursive comprehension, which contradicts that f is unique.

Apply $\mathsf{Ext}(\omega^*)$ to the well-founded partial order $(A, \leq'_A)$ to obtain a well-order $(A, \leq_A)$. This well-order $(A, \leq_A)$ is as desired. If not, then there is some $a \in A$ with $R(a) - R(i(a)) = \{f(a), b\}$. Note b cannot be $f(a')$ for any other $a' \in A$, since that would imply $a' \leq a$ and $b \in R(i(a))$. So we may similarly define a second solution by letting g agree with f everywhere except a, and setting $g(a) = b$. As $\mathsf{Ext}(\omega^*)$ follows from ACA_0, the proof is complete. $\square$

While we cannot reverse STO to ACA_0, we obtain a partial reversal to the principle $\mathsf{Ext}(\omega^*)$.

Theorem 2.2.4. $\mathsf{RCA}_0 + \mathsf{STO} \vdash \mathsf{Ext}(\omega^*)$.

Proof. Let P be a well-founded partial order. Without loss of generality, assume $P = (\mathbb{N}, \leq_N)$. We define a matching problem $P' = (\mathbb{N}, \mathbb{N}, R)$ with

$$R = \{(n, m) : m \leq_P n\}.$$

Note for each n, $R(n) = \{m : m \leq_P n\}$. Clearly, the identity map $f : \mathbb{N} \to \mathbb{N}$ is a solution to P', and both P' and f exist by recursive comprehension.

We claim that f is unique. Suppose not and let $g : \mathbb{N} \to \mathbb{N}$ be a solution differing from f. Fix an n_0 such that $g(n_0) \neq f(n_0) = n_0$. Apply primitive recursion to define the sequence $\langle n_i \rangle_{i \in \mathbb{N}}$ with $n_{i+1} = g(n_i)$. Note for each i, $n_{i+1} \leq_P n_i$ as $g(n_i) = n_{i+1}$. Because g is injective, it follows from Σ^0_0 induction that $n_{i+1} \neq n_i$ for all i. Thus P

contains an infinite descending chain $n_0 >_P n_1 >_P \cdots$, a contradiction. Apply STO to P' and obtain a well order $(\mathbb{N}, \leq')$ such that for all n, there exists a unique m with

$$R(n) - R(i_{\leq'}(n)) = \{m\}.$$

We need to show that $(\mathbb{N}, \leq')$ extends $(\mathbb{N}, \leq_P)$, i.e., for every n and m, if $n \leq_P m$ then $n \leq' m$. Call n an *error witnessed by* m if $n <' m$, but $m <_P n$. Thus it suffices to show that no n is an error. By way of contradiction, suppose not: then there exists some pair n and m such that n is an error witnessed by m. Notice then that m must be an error as well. Indeed, as $n \in i_{\leq'}(m)$ and $m \in R(n)$, $R(m) - R(i_{\leq'}(m)) = \{\ell\}$ for some $\ell \neq m$. Note $\ell <_P m$ because $\ell \in R(m)$, and since $\ell \notin R(i_{\leq'}(m))$ we have $m <' \ell$. Thus ℓ witnesses that m is an error.

Note also that n being an error is definable by a Σ^0_1 formula: there exists a witness to show n is an error. If follows then from Lemma II.3.7 of Simpson [22] that there is either a finite set X containing all errors, or there is an injection $h : \mathbb{N} \to \mathbb{N}$ such that n is an error if and only if there is an i such that $h(i) = n$. We show neither case can obtain, as else we reach a contradiction. For the first case, let X be the set of all errors, and let n be the $\leq'$-least element of X. Let m witness that n is an error. Then for all $\ell <' n$, ℓ cannot be an error. So $n \not\leq_P \ell$ and $m \not\leq_P \ell$ because $\ell <' n <' m$. Hence $m, n \notin R(\ell)$ for any $\ell <' n$, and so $R(n) - R(i_{\leq'}(n)) = \{n, m\}$, a contradiction.

For the latter case, we use h to define an infinite descending chain in P. First, define an auxiliary function $h_1(n, m)$ which outputs 1 if n is an error witnessed by

$h(m)$. Specifically, set

$$h_1(n,m) = \begin{cases} 1 & \text{if } n <' h(m) \text{ and } h(m) <_P n \\ 0 & \text{otherwise} \end{cases}$$

Note if n is an error, then such an m must exist as any witness for n is also an error. By minimization, there exists a function j such that $j(n)$ equals the least m with $h_1(n,m) = 1$. Intuitively, $j(n)$ is the least index (with respect to h) of an error witnessing that n itself is an error. Apply primitive recursion to define the sequence $\langle n_i \rangle_{i\in\mathbb{N}}$ given by $n_0 = h(0)$ and $n_{i+1} = h(j(n_i))$. Notice for each i, $n_{i+1} <_P n_i$ because $h_1(n_i, j(n_i)) = 1$ which ensures $n_{i+1} = h(j(n_i)) <_P n_i$. Thus $\langle n_i \rangle_{i\in\mathbb{N}}$ is an infinite descending chain in P, contradicting that P is well-founded.

We conclude P contains no errors which completes the proof. □

Corollary 2.2.5. *The statement* STO *is not provable in* WKL_0.

Proof. By Theorem 1 of [5], $\mathsf{Ext}(\omega^*)$ is not provable in WKL_0. By Theorem 2.2.4, $\mathsf{Ext}(\omega^*)$ is provable in $\mathsf{RCA}_0 + \mathsf{STO}$. Hence, STO is not provable in WKL_0. □

2.3 Variants of STO in reverse mathematics

As mentioned above, we do not currently know of any reversal of STO to ACA_0. We conjecture one exists.

Conjecture 2.3.1 (RCA_0). *The following are equivalent*

1. ACA_0

2. STO

The difficulty in obtaining ACA_0 from STO seems to arise due to the relatively weak coding potential in the arbitrary well-order STO yields. In view of this difficulty, we weaken the principle STO to possibly illuminate under what assumptions these difficulties might vanish. As mentioned above, a principle like STO was studied in Hirst and Hughes [18] for matching problems obeying Hall's condition. Specifically, the following analogue of STO appears in Theorem 7 of [18].

Theorem 2.3.2. *Over* RCA_0*, the following are equivalent:*

1. ACA_0

2. *Suppose $P = (A, B, R)$ is a matching problem such that every $a \in A$ has only finitely many permissible matches. If P has a unique solution, then there is an enumeration $\langle a_i \rangle_{i \geq 1}$ of A such that $|R(a_1, \ldots, a_n)| = n$ for every $n \geq 1$.*

Contrasting item 2 with STO yields two key differences. The first is, of course, Hall's condition, but the second and key difference is the structure of the resulting object guaranteed by the principle. Here, the resulting object is a *sequence*, while STO only guarantees a well-order. Obtaining a sequence yields two advantages. First, the order type of the underlying well-order is ω; second, there is an effective way to determine the set of predecessors of any element in the sequence. Neither of these may hold in the well-order which STO yields. Thus, we are motivated to investigate the following weakenings of STO.

Statement 2.3.3. $\mathsf{STO(F)}$: Let $P = (A, B, R)$ be a matching problem with a unique solution *in which every element has finitely many permissible matches.* Then there is a well-order $(A, \leq_A)$ such that for every $a \in A$, there is a unique $b \in B$ such that

$$R(a) - R(i(a)) = \{b\}.$$

Statement 2.3.4. $\mathsf{STO}(\omega)$: Let $P = (A, B, R)$ be a matching problem with a unique solution, in which every element has finitely many permissible matches, and A is infinite. Then there is a well-order $(A, \leq_A)$ *of type* ω such that for every $a \in A$, there is a unique $b \in B$ such that

$$R(a) - R(i(a)) = \{b\}.$$

So $\mathsf{STO}(\mathsf{F})$ is STO with Hall's condition reinstated, and $\mathsf{STO}(\omega)$ forces every element to have finitely many predecessors in the resulting well-order. Based on the similarity of these principles to item 2 in Theorem 2.3.2, it is reasonable to suspect them provably equivalent to ACA_0 over RCA_0. As we show next, this is indeed the case with $\mathsf{STO}(\omega)$, but the proof requires a completely new reversal than what appears in Hirst and Hughes [18].

Theorem 2.3.5 (RCA_0)**.** *The following are equivalent*

1. ACA_0
2. $\mathsf{STO}(\omega)$

Proof. To see that (1) implies (2), let $P = (A, B, R)$ be a matching problem which satisfies Hall's condition and has a unique solution. Apply Theorem 2.3.2 to obtain an enumeration of A, $\langle a_i \rangle_{i \geq 1}$ such that $|R(a_1, \ldots, a_n)| = n$ for every $n \geq 1$. Clearly, the well-order $(A, \leq_A)$ defined by $a_m \leq_A a_n$ if and only if $m \leq n$ is as desired.

For the reversal, we work in RCA_0 and use $\mathsf{STO}(\omega)$ to deduce the contrapositive of Kőnig's lemma. This suffices by Theorem III.7.2 of Simpson [22]. Let $T \subset \mathbb{N}^{<\mathbb{N}}$ be a finitely branching tree with no infinite paths. We show T must be finite. Assume for sake of contradiction that T is infinite. Let $A = B = T$ and define $R \subseteq A \times B$ as follows

$$(\sigma, \tau) \in R \iff (\sigma = \tau) \vee (|\tau| = |\sigma| + 1 \wedge \sigma \preceq \tau).$$

Let $P = (A, B, R)$ be the associated matching problem, and note that P is computable in T, and hence exists by Δ^0_1 comprehension. Note that as T is finitely branching, for each $\sigma \in A$, $R(\sigma)$ is finite as σ has finitely many immediate successors in T. Thus P satisfies Hall's condition.

We claim P has as a unique solution, namely the identity function f. Clearly, f exists by Δ^0_1 comprehension and is a solution. To see that it is unique, suppose g is a distinct solution, and let σ_0 be such that $\sigma_0 = f(\sigma_0) \neq g(\sigma_0)$. Let $\sigma_1 = g(\sigma_0)$, and note by construction that σ_1 is an immediate successor of σ_0 and as g is a solution, $g(\sigma_1) \neq \sigma_1 = g(\sigma_0)$. Proceeding by induction, we define $\sigma_n = g(\sigma_{n-1})$; note that for each n, σ_n is an immediate successor of σ_{n-1}. Hence closing $\{\sigma_n : n \in \mathbb{N}\}$ under prefix defines an infinite path in T, a contradiction.

Apply $\mathsf{STO}(\omega)$ to obtain a well-order of $A = T$ of type ω such that for each $\sigma \in A$, there is a unique $\tau \in B$ with

$$R(\sigma) - R(i(\sigma)) = \{\tau\}.$$

We claim that if $\sigma \preceq \tau$ then $\tau \leq_A \sigma$. To see this, suppose not and let τ be the lexicographically least string such that there is a σ with $\sigma \prec \tau$ but $\sigma \leq_A \tau$. Note as τ

is least, there can be no string $\sigma \prec \mu \prec \tau$ with $\tau \leq_A \mu$ by transitivity. Thus τ is an immediate successor of σ. Note that σ is the only element of T not equal to τ with $\tau \in R(\sigma)$. Again, as τ is least, all $\mu \in T$ with $\mu \preceq \sigma$ must appear later in $\leq_A$ than σ, that is $\sigma \leq_A \mu$. This implies that every element $\mu \leq_A \sigma$ satisfies

$$R(\mu) \cap \{\sigma, \tau\} = \emptyset.$$

Hence, $R(\sigma) - R(i_{\leq_A}(\sigma)) = \{\sigma, \tau\}$, a contradiction. This verifies the claim.

Now, every element of T appears before λ in $(A, \leq_A)$. Since this well-order is of type ω, λ must have only finitely many $\leq_A$-predecessors. So T is finite and this completes the proof. □

If one could somehow refine an instance $P = (A, B, R)$ of STO to an instance $P' = (A, B, R')$ satisfying Hall's condition in such a way that an application of $\mathsf{STO}(\omega)$ correctly orders A with respect to the original relation R, we would obtain a reversal to ACA_0. This motivates an analysis of these principles using computable and Weihrauch reductions to determine if this is possible. We conjecture STO and $\mathsf{STO}(\omega)$ are not equivalent under one of $\leq_{\mathsf{c}}$, $\leq_{\mathsf{sc}}$, $\leq_{\mathsf{W}}$ or $\leq_{\mathsf{sW}}$.

Now, turning our attention to $\mathsf{STO}(\mathsf{F})$, we obtain immediately from Theorem 2.3.5 the following.

Corollary 2.3.6. *The statement* $\mathsf{STO}(\mathsf{F})$ *is provable in* ACA_0.

However, losing the guarantee that the well-order is of type ω reintroduces the original coding difficulties with STO. Though we gain Hall's condition as a hypothesis in $\mathsf{STO}(\mathsf{F})$, we so far are only able to show $\mathsf{STO}(\mathsf{F})$ implies WKL_0 over RCA_0.

The idea of the proof is to use a well-order to guide the way up an infinite path in a subtree T of $2^{<\mathbb{N}}$. Specifically, we will build a matching problem $(\mathcal{A}, \mathcal{B}, R)$ with pairs $a_\sigma, b_\sigma \in \mathcal{A}$ for each $\sigma \in T$. The resulting well-order $(\mathcal{A}, \leq_{\mathcal{A}})$ will then have $a_\sigma <_{\mathcal{A}} b_\sigma$ if $\sigma^\frown 0$ is extendible to an infinite path, or $b_\sigma <_{\mathcal{A}} a_\sigma$ if $\sigma^\frown 1$ is extendible to an infinite path. In the event that either is extendible, the well-order will make the arbitrary choice of which side to follow.

The way we will achieve this is with auxiliary elements $c, d \in \mathcal{A}$. To illustrate the idea notice if $(a, b), (b, a), (c, d), (d, c) \in R$ then matching a and b, and c and d, will result in a solution. To force $a <_{\mathcal{A}} b$, we further add $(c, b), (b, d)$ to R. Then a must come before c, and c must come before b. Similarly, we can ensure $b <_{\mathcal{A}} a$ by adding (c, a) and (a, d) to R. This decision will be made when we see a certain string τ either has $\tau^\frown 0 \notin T$ or $\tau^\frown 1 \notin T$.

Theorem 2.3.7. *Over* RCA_0*,* $\mathsf{STO(F)}$ *implies* WKL_0

Proof. Let T be an arbitrary infinite tree in $2^{<\mathbb{N}}$. We build a matching problem $P = (\mathcal{A}, \mathcal{B}, R)$ computable in T as follows. We first construct five countable sequences of sets $\langle A_n \rangle$, $\langle B_n \rangle$, $\langle C_n \rangle$, $\langle D_n \rangle$ and $\langle R_n \rangle$. We conclude by setting

$$A = \bigcup_{n \in \mathbb{N}} A_n, \quad B = \bigcup_{n \in \mathbb{N}} B_n, \quad C = \bigcup_{n \in \mathbb{N}} C_n, \quad D = \bigcup_{n \in \mathbb{N}} D_n, \quad R = \bigcup_{n \in \mathbb{N}} R_n,$$

and $\mathcal{A} = \mathcal{B} = A \cup B \cup C \cup D$.

Construction. To begin, let $A' = \{a_0, a_1, \dots\}$, $B' = \{b_0, b_1, \dots\}$, $C' = \{c_0, c_1, \dots\}$, and $D' = \{d_0, d_1, \dots\}$ be four infinite disjoint computable subsets of $\mathbb{N}$. We pull from these sets to build the constituents of P. For ease of notation, let $a_\sigma = a_m$ where m is the code for $\sigma \in T$. Define b_σ, c_σ and d_σ similarly. To begin the construction, let $A_{-1} = \{a_\lambda\}$, $B_{-1} = \{b_\lambda\}$, and $C_{-1} = D_{-1} = \emptyset$. Let $R_{-1} = \{(a_\lambda, b_\lambda)\}$.

Distribute the stages of the construction so that every string $\tau \in 2^{<\omega}$ of length n is considered in turn. Suppose we are at stage n and considering string τ. Unless explicitly defined otherwise, we set $A_n = A_{n-1}$, $B_n = B_{n-1}$, $C_n = C_{n-1}, D_n = D_{n-1}$, and $R_n = R_{n-1}$.

If $\tau \notin T$, proceed to the next stage. If $\tau \in T$, determine if $\tau^\frown 0 \in T$, and if $\tau^\frown 1 \in T$. This leads to four cases. After we define R_n, the stage is complete.

Case 1: $\tau^\frown 0 \in T$ and $\tau^\frown 1 \in T$. As both successors of τ are in T, either may be extendible to a path, so we add a_τ and b_τ to the problem without forcing either of $a_\tau <_{\mathcal{A}} b_\tau$ or $b_\tau <_{\mathcal{A}} a_\tau$. Specifically, set $A_n = A_{n-1} \cup \{a_\tau\}$, $B_n = B_{n-1} \cup \{b_\tau\}$, and

$$R_n = R_{n-1} \cup \{(a_\tau, b_\tau), (b_\tau, a_\tau)\}.$$

Case 2: $\tau^\frown 0 \in T$, but $\tau^\frown 1 \notin T$. In this case, as τ is not extendible via $\tau^\frown 1$, we ensure the well-order puts $a_\tau <_{\mathcal{A}} b_\tau$. To do this, let

$$A_n = A_{n-1} \cup \{a_\tau\},\ B_n = B_{n-1} \cup \{b_\tau\},\ C_n = C_{n-1} \cup \{c_\tau\} \text{ and } D_n = D_{n-1} \cup \{d_\tau\},$$

and set

$$R_n = R_{n-1} \cup \{(a_\tau, b_\tau), (b_\tau, a_\tau)\} \cup \{(c_\tau, d_\tau), (d_\tau, c_\tau)\} \cup \{(c_\tau, b_\tau), (b_\tau, d_\tau)\}.$$

Case 3: $\tau^\frown 0 \notin T$ and $\tau^\frown 1 \in T$. Here we ensure the well-order has $b_\tau <_{\mathcal{A}} a_\tau$. To do this, update A_n, B_n, C_n, and D_n as in case 2. Set

$$R_n = R_{n-1} \cup \{(a_\tau, b_\tau), (b_\tau, a_\tau)\} \cup \{(c_\tau, d_\tau), (d_\tau, c_\tau)\} \cup \{(c_\tau, a_\tau), (a_\tau, d_\tau)\}.$$

Case 4: $\tau^\frown 0 \notin T$ and $\tau^\frown 1 \notin T$. In this case τ is not extendible to an infinite path in T. And moreover, for any predecessor σ of τ, one of $\sigma^\frown 0$ or $\sigma^\frown 1$ is also not extendible. We find the longest possible string σ for which this has not already been encoded into the matching problem and update R_n with respect to this σ.

Specifically, set $C_n = C_{n-1} \cup \{c_\tau\}$, $D_n = D_{n-1} \cup \{d_\tau\}$, and fix the string $\sigma \prec \tau$ of greatest length which has a successor of length $n+1$ in T. Such a σ is guaranteed as T is infinite. Let υ be the witnessing successor of σ. So $\upsilon \restriction |\sigma| = \tau \restriction |\sigma| = \sigma$, but $\upsilon(|\sigma|) \neq \tau(|\sigma|)$. If $\upsilon(|\sigma|) = 0$, then $\tau(|\sigma|) = 1$, so we ensure $a_\sigma <_{\mathcal{A}} b_\sigma$. Set

$$R_n = R_{n-1} \cup \{(c_\tau, d_\tau), (d_\tau, c_\tau)\} \cup \{(c_\tau, b_\sigma), (b_\sigma, d_\tau)\}.$$

If instead $\upsilon(|\sigma|) = 1$, we have $\tau(|\sigma|) = 0$, so we make $b_\sigma <_{\mathcal{A}} a_\sigma$. To do this, set

$$R_n = R_{n-1} \cup \{(c_\tau, d_\tau), (d_\tau, c_\tau)\} \cup \{(c_\tau, a_\sigma), (a_\sigma, d_\tau)\}.$$

This completes the construction.

Verification. We need show that P has a unique solution, satisfies Hall condition, and that any well-order guaranteed by $\mathsf{STO(F)}$ encodes a path in T. We do each in succession.

Claim. *The map $f : \mathcal{A} \to \mathcal{B}$ defined by*

$$a_\sigma \mapsto b_\sigma, \quad b_\sigma \mapsto a_\sigma, \quad c_\sigma \mapsto d_\sigma, \text{ and } d_\sigma \mapsto c_\sigma,$$

for all $\sigma \in T$ is the unique solution of P.

Clearly f is a solution to P. To show that f is unique, we first show that no string

$\sigma \in T$ gave rise to a_σ and b_σ which were each forced to have $a_\sigma <_{\mathcal{A}} b_\sigma$ and $b_\sigma <_{\mathcal{A}} a_\sigma$. That is $R(a_\sigma) \neq \{b_\sigma\}$ only if $R(b_\sigma) = \{a_\sigma\}$, and $R(b_\sigma) \neq \{a_\sigma\}$ only if $R(a_\sigma) = \{b_\sigma\}$.

Suppose for sake of contradiction that this occurred. Then there are distinct successors of σ in T, say τ and υ, such that

$$\{(c_\tau, a_\sigma), (a_\sigma, d_\tau), (c_\upsilon, b_\sigma), (b_\sigma, d_\upsilon)\} \subset R$$

This implies that σ is the longest substring of τ with a successor of length $|\tau| + 1$ in T. Similarly, σ is the longest substring of υ with a successor of length $|\upsilon| + 1$ in T. Without loss of generality, assume $|\upsilon| \leq |\tau|$. By construction, we must have that $\tau(|\sigma|) = 1$ and $\upsilon(|\sigma|) = 0$. In particular, we have that $\tau(|\sigma|) \neq \upsilon(|\sigma|)$. So it must be the case that $\sigma^\frown 0$ has a successor in T of length $|\tau| + 1$. In particular, $\sigma^\frown 0$ must have a successor T of length $|\upsilon| + 1 \leq |\tau| + 1$. This contradicts the fact that σ is the longest predecessor of υ with a successor of length $|\upsilon| + 1$ for $\sigma^\frown 0 \preceq \upsilon$ and has length greater that $|\sigma|$.

Hence we see if $R(a_\sigma) \neq \{b_\sigma\}$, then $R(b_\sigma) = \{a_\sigma\}$, and if $R(b_\sigma) \neq \{a_\sigma\}$, then $R(a_\sigma) = \{b_\sigma\}$. This allows us to show that f is unique. Suppose toward a contradiction that h is a distinct solution of P. Fix the lexicographically least $\tau \in T$ such that $h(c_\tau) \neq f(c_\tau) = d_\tau$. Note if h disagrees on any a_τ or b_τ, it must disagree on some c_τ as well. There must be some $\sigma \in T$ such that $h(c_\tau) \in \{a_\sigma, b_\sigma\}$. Without loss of generality, assume $h(c_\tau) = a_\sigma$. Then $h(b_\sigma) \neq a_\sigma$ as h is an injection. So $R(b_\sigma) = \{a_\sigma, h(b_\sigma)\}$ which implies $R(a_\sigma) = \{b_\sigma\}$. But by the construction, we must have that $d_\tau \in R(a_\sigma)$ since $a_\sigma \in R(c_\tau)$, a contradiction.

This verifies the claim.

Claim. *The problem $P = (\mathcal{A}, \mathcal{B}, R)$ satisfies Hall's condition. That is, for every*

$x \in \mathcal{A}$, the set $R(x)$ is finite.

For sake of contradiction, suppose some element of $\mathcal{A}$ has infinitely many permissible matches. This must be an element of A or B, since for all $n \in \omega$, $|R(c_n)| \leq 2$ and $|R(d_n)| \leq 2$. Without loss of generality, assume the element to be a_σ for some $\sigma \in T$. Specifically, suppose

$$R(a_\sigma) = \{b_\sigma, d_{\tau_0}, d_{\tau_1}, d_{\tau_2}, \dots\}.$$

By construction we must have for every i, that $\sigma^\frown 1 \preceq \tau_i$; that σ is the longest predecessor of τ_i such that $\sigma^\frown 0$ has a successor of length $|\tau_i| + 1$; and that no τ_i has a successor in T. Take some τ_j and τ_k for which $|\tau_j| < |\tau_k|$. Note $\tau_j \not\prec \tau_k$ but $\sigma^\frown 1 \prec \tau_j$ and $\sigma^\frown 1 \prec \tau_k$. Thus $\sigma^\frown 1$ has a successor of length $|\tau_k| \geq |\tau_j| + 1$. This contradicts that σ was the longest predecessor of τ_j with a successor of length $|\tau_j| + 1$. Thus $R(n)$ is finite for all $n \in \mathcal{A}$.

Claim. *Any well-order on $\mathcal{A}$ satisfying the conclusion of* $\mathsf{STO}(\mathsf{F})$ *computes a path in T.*

As P is a matching problem with a unique solution in which every element has finitely many permissible matches, we apply $\mathsf{STO}(\mathsf{F})$ to obtain a well-order $(\mathcal{A}, \leq_{\mathcal{A}})$ such that for every $x \in A$, there is a unique $y \in \mathcal{B}$ such that

$$R(x) - R(i_{\leq_\mathcal{A}}(x)) = \{y\}.$$

Define a sequence $g = \langle g(n) \rangle_{n \in \mathbb{N}}$ recursively as follows:

$$g(0) = \begin{cases} 0 & \text{if } a_\lambda \leq_A b_\lambda \\ 1 & \text{if } b_\lambda \leq_A a_\lambda \end{cases} \text{ and } g(n) = \begin{cases} 0 & \text{if } a_{g\restriction n} \leq_A b_{g\restriction n} \\ 1 & \text{if } b_{g\restriction n} \leq_A a_{g\restriction n} \end{cases}.$$

Clearly g is computable in $\leq_{\mathcal{A}}$. We claim g is a path in $[T]$.

Clearly $g \upharpoonright 0 = \lambda$ has infinitely many successors in T because T is infinite. Suppose by induction that $g \upharpoonright n$ has infinitely many successors in T. If both $(g \upharpoonright n)^\frown 0$ and $(g \upharpoonright n)^\frown 1$ have infinitely many successors in T, then so must $g \upharpoonright (n+1)$. Suppose without loss of generality that $(g \upharpoonright n)^\frown 0$ does not. Then by our induction hypothesis $(g \upharpoonright n)^\frown 1$ must have infinitely many successors. Moreover there is a maximum length ℓ such that if $\tau \in T$ with $(g \upharpoonright n)^\frown 0 \preceq \tau$, then $|\tau| \leq \ell$. Hence, there is some $\tau \succeq (g \upharpoonright n)^\frown 0$ which has no successors in T, such that at a stage in our construction we ensured

$$\{(c_\tau, d_\tau)\} \cup \{(c_\tau, a_{(g\upharpoonright n)}), (a_{(g\upharpoonright n)}, d_\tau)\} \subseteq R.$$

Thus we must have $b_{(g\upharpoonright n)} \leq_{\mathcal{A}} c_\tau \leq_{\mathcal{A}} a_{(g\upharpoonright n)}$. The definition of g then implies that we have $g(n) = 1$. So $g \upharpoonright (n+1) = (g \upharpoonright n)^\frown 1$ has infinitely many successors in T.

We conclude that $g \upharpoonright n \in T$ for all n. Thus $g \in [T]$ and the proof is complete. □

We conjecture that WKL_0 is not sufficient to prove $\mathsf{STO(F)}$. A reversal of $\mathsf{STO(F)}$ to ACA_0 would yield a reversal for STO.

We conclude by weakening STO sufficiently to obtain a principle provable in WKL_0.

Definition 2.3.8. We say a matching problem $P = (A, B, R)$ is *bounded* if there is a function $h : A \to \mathbb{N}$ such that for all $a \in A$, if $b \in R(a)$, then $b < h(a)$.

Statement 2.3.9. $\mathsf{STO(B)}$: If $P = (A, B, R)$ is a *bounded* matching problem with a unique solution, then there is a well-order $(A, \leq_A)$ such that for every $a \in A$, there is a unique $b \in B$ such that

$$R(a) - R(i(a)) = \{b\}.$$

Bounded matching problems were studied in Hirst [15] and Hirst and Hughes [18].

In particular, Theorem 9 of [18] shows that the restriction of Item 2 from Theorem 2.3.2 to bounded matching problems is equivalent to WKL_0 over RCA_0. We state it here for convenience.

Theorem 2.3.10. *Over RCA_0, the following are equivalent:*

1. WKL_0
2. *Suppose $P = (A, B, R)$ is a bounded matching problem. If P has a unique solution, then there is an enumeration $\langle a_i \rangle_{i \geq 1}$ of A such that $|R(a_1, \ldots, a_n)| = n$ for every $n \geq 1$.*

Corollary 2.3.11. *The statement $\mathsf{STO(B)}$ is provable in WKL_0.*

Though this fact follows immediately from Theorem 2.3.10, the reversal used in [18] does not yield a reversal of $\mathsf{STO(B)}$ to WKL_0. Indeed, the enumeration guaranteed by Theorem 2.3.10 is again paramount in that result. It remains open if $\mathsf{STO(B)}$ implies WKL_0 over RCA_0.

As the original difficulty of encoding sufficient information in an well-order remains in each variant of STO, we wonder if indeed STO, $\mathsf{STO(F)}$, or $\mathsf{STO(B)}$ occupy some area in the reverse mathematics zoo distinct from WKL_0 and ACA_0. Contrasting these statements with other principles therein will be the subject of future work, alongside a treatment of these principles under computable and Weihrauch reductions. We are confident that a reversal of any one of these principles would shed light on the other open implications.

Chapter 3

Finding chains and antichains in ω-ordered posets

In this chapter we introduce and investigate $\mathsf{CAC}^{\mathsf{ord}}$, a variant of the *chain-antichain principle*. The chain-antichain principle CAC states that any infinite poset $(P, \leq_P)$ contains either an infinite chain or an infinite antichain. The principle $\mathsf{CAC}^{\mathsf{ord}}$ is then CAC restricted to *ω-ordered* posets: i.e. a poset $(P, \leq_P)$ such that if $x \leq_P y$ then $x < y$. We also study the stable versions $\mathsf{SCAC}^{\mathsf{ord}}$ and SCAC, which are respectively $\mathsf{CAC}^{\mathsf{ord}}$ and CAC restricted to *stable posets* as defined by Hirschfeldt and Shore [13]. We show that while CAC and $\mathsf{CAC}^{\mathsf{ord}}$, and SCAC and $\mathsf{SCAC}^{\mathsf{ord}}$, are equivalent in reverse mathematics, they are not equivalent in the view of computable and Weihrauch reductions. In particular, we show that $\mathsf{CAC}^{\mathsf{ord}} \not\leq_{\mathsf{c}} \mathsf{CAC}$, $\mathsf{SCAC}^{\mathsf{ord}} \not\leq_{\mathsf{W}} \mathsf{SCAC}$ and $\mathsf{SCAC}^{\mathsf{ord}} \not\leq_{\mathsf{sc}} \mathsf{SCAC}$ and that these non-reductions are sharp. The first is obtained by analyzing the complexity of infinite chains and infinite antichains in ω-ordered posets, while the latter two results are obtained respectively by a Seetapun-style finite extension argument and by a forcing construction which utilizes nested applications of the tree labeling technique.

3.1 Definitions and preliminary results

Recall a partially ordered set or *poset* is any pair $(P, \leq_P)$ where $\leq_P$ is a reflexive, transitive, and antisymmetric relation on P. We call P and $\leq_P$ respectively the *domain* and *partial order* of $(P, \leq_P)$. All posets considered herein have domain contained in ω. A *chain* C in $(P, \leq_P)$ is a subset of P such that any two elements in C are comparable under $\leq_P$, i.e.,

$$x, y \in C \to (x \leq_P y \vee y \leq_P x).$$

An *antichain* A in $(P, \leq_P)$ is a subset of P in which any two distinct elements are incomparable under $\leq_P$, written $x \mid_P y$. If $Q \subseteq P$, we use $(Q, \leq_P^Q)$ to denote the suborder of P given by $\leq_P$ restricted to elements in Q. Thus

$$(Q, \leq_P^Q) = (Q, \leq_P \restriction (Q \times Q)).$$

The *chain-antichain principle* CAC asserts that any infinite poset must contain an infinite chain or an infinite antichain. This can be viewed as an infinitary version of Dilworth's Theorem which connects the size of a maximal antichain in a finite poset to the number of chains in that poset. Hirschfeldt and Shore [13] initialized the study of CAC in reverse mathematics and introduced a *stable* version denoted SCAC.

Definition 3.1.1. For an infinite poset $(P, \leq_P)$ we say an element $x \in P$ is

- *small* if $x \leq_P y$ for all but finitely many $y \in P$;
- *large* if $y \leq_P x$ for all but finitely many $y \in P$;

- *isolated* if $x \mid_P y$ for all but finitely many $y \in P$.

We say P is *stable* if all elements are either small or isolated, or all elements are either large or isolated. In the first case, we call P a stable poset of the *small type*. In the second case, we call P a stable poset of the *large type*

Statement 3.1.2. SCAC: If $(P, \leq_P)$ is an infinite stable poset, then P contains either an infinite chain or an infinite antichain.

Frequently, the posets constructed in the use and study of principles like CAC and SCAC are what we call ω-*ordered*. That is, the poset $(P, \leq_P)$ respects the natural order of ω: $x \leq_P y$ only if $x \leq y$.

Definition 3.1.3. We say that poset $(P, \leq_P)$ with $P \subseteq \mathbb{N}$ is ω-*ordered* if for all $x, y \in P$, $x \leq_P y$ implies that $x \leq y$.

This motivates the study of the principles $\mathsf{CAC}^{\mathsf{ord}}$ and $\mathsf{SCAC}^{\mathsf{ord}}$ which are, respectively, CAC and SCAC restricted to ω-ordered posets.

Statement 3.1.4. Let $(P, \leq_P)$ be an infinite poset. The principles $\mathsf{CAC}^{\mathsf{ord}}$ and $\mathsf{SCAC}^{\mathsf{ord}}$ are as follows:

$\mathsf{CAC}^{\mathsf{ord}}$: If $(P, \leq_P)$ is ω-ordered, then P contains an infinite chain or an infinite antichain.

$\mathsf{SCAC}^{\mathsf{ord}}$: If $(P, \leq_P)$ is stable and ω-ordered, then P contains an infinite chain or an infinite antichain.

For every principle given in this section, we formulate them as *problems* for computable and Weihrauch inductions by defining *instances* as posets which satisfy the required hypotheses paired with *solutions* that are either an infinite chain or an infinite antichain.

To begin our analysis, we show that CAC and $\mathsf{CAC}^{\mathsf{ord}}$ are equivalent in the sense of reverse mathematics. That CAC and $\mathsf{CAC}^{\mathsf{ord}}$ are equivalent over ω-models is implicit in Lemma 2.12 of Towsner [24].

Theorem 3.1.5. *Over* RCA_0*, the principles* CAC *and* $\mathsf{CAC}^{\mathsf{ord}}$ *are equivalent.*

Proof. Proving $\mathsf{CAC}^{\mathsf{ord}}$ from CAC in RCA_0 is trivial. For the other direction assume $\mathsf{CAC}^{\mathsf{ord}}$ and work in RCA_0. Let $(P, \leq_P)$ be an infinite poset. We refine $(P, \leq_P)$ to an ω-ordered poset and use $\mathsf{CAC}^{\mathsf{ord}}$ to find an infinite chain or antichain Y of $(P, \leq_P)$

Note the following sets are Δ^0_1-definable over $(P, \leq_P)$:

$$\leq_+ = \{(x, y) : x \leq_P y \wedge x \leq y\} \text{ and } \leq_- = \{(y, x) : x \leq_P y \wedge y \leq x\}.$$

Hence RCA_0 proves that $(P, \leq_+)$ and $(P, \leq_-)$ are infinite ω-ordered posets.

Apply $\mathsf{CAC}^{\mathsf{ord}}$ to $(P, \leq_+)$ to obtain an infinite set X which is a chain or antichain with respect to $\leq_+$. If X is a chain, then it is a chain under $\leq_P$ and we are done. Assume instead that X is an antichain in $\leq_+$, and note it may not be an antichain under $\leq_P$. Consider $(P, \leq_-)$ restricted to X: this poset $(X, \leq^X_-)$ is infinite and ω-ordered. Apply $\mathsf{CAC}^{\mathsf{ord}}$ to obtain an infinite set $Y \subseteq X$ which is either a chain or antichain with respect to $\leq_-$. In either case, we claim Y is an infinite chain or infinite antichain of $\leq_P$.

If Y is a chain in $\leq_-$, then it is an infinite chain with respect to $\leq_P$ in the opposite direction. That is

$$x, y \in Y \rightarrow (x \leq_- y) \rightarrow (y \leq_P x).$$

If Y is instead an antichain in $\leq_P^-$, then as $Y \subseteq X$, no pair in Y is comparable in either $\leq_-$ or $\leq_+$. Hence Y is an infinite antichain in $\leq_P$ and the claim is verified. This completes the proof. $\square$

Notice the above proof can be carried out uniformly in the given poset $(P, \leq_P)$. The first application of $\mathsf{CAC}^{\mathsf{ord}}$ to a computable suborder of $\leq_P$ yields an infinite set X. Independent of whether or not X is a chain or antichain, a second application of $\mathsf{CAC}^{\mathsf{ord}}$ to a computable suborder of $\leq_P$ restricted to X yields an infinite set Y which must be an infinite chain or infinite antichain in $(P, \leq_P)$. We may introduce a non-uniform decision to not apply $\mathsf{CAC}^{\mathsf{ord}}$ a second time in the event that X is a chain, but this is unnecessary. Hence we obtain the following.

Theorem 3.1.6. $\mathsf{CAC} \equiv_{\mathsf{gW}} \mathsf{CAC}^{\mathsf{ord}}$.

Proof. To see $\mathsf{CAC} \leq_{\mathsf{gW}} \mathsf{CAC}^{\mathsf{ord}}$, consider the following strategy for $G(\mathsf{CAC}^{\mathsf{ord}} \to \mathsf{CAC})$. Given $(P, \leq_P)$, an instance of CAC, compute $\leq_+$ and play the infinite ω-ordered poset $(P, \leq_+)$. Now a solution to $(P, \leq_+)$ is an infinite subset $X \subseteq P$. Given X and $(P, \leq_P)$, compute and play the infinite ω-ordered poset$(X, \leq_-^X)$. Any solution of $(X, \leq_-^X)$ is identically a solution to $(P, \leq_P)$ by the proof of Theorem 3.1.5. Thus, this strategy is winning.

The reverse direction is trivial as every instance of $\mathsf{CAC}^{\mathsf{ord}}$ is an instance of CAC. Hence $\mathsf{CAC}^{\mathsf{ord}} \leq_{\mathsf{sW}} \mathsf{CAC}$ and in particular $\mathsf{CAC}^{\mathsf{ord}} \leq_{\mathsf{gW}} \mathsf{CAC}$. $\square$

Though the proof of Theorem 3.1.5 is uniform, it does require two uses of $\mathsf{CAC}^{\mathsf{ord}}$ in series. This suggests that CAC is not Weihrauch reducible to $\mathsf{CAC}^{\mathsf{ord}}$. Indeed, showing $\mathsf{CAC} \not\leq_{\mathsf{W}} \mathsf{CAC}^{\mathsf{ord}}$ would prove that multiple uses of $\mathsf{CAC}^{\mathsf{ord}}$ are necessary in any proof of $\mathsf{CAC}^{\mathsf{ord}} \to \mathsf{CAC}$ over a weak enough base system. We obtain this fact by a sharper result, namely that $\mathsf{CAC} \not\leq_{\mathsf{c}} \mathsf{CAC}^{\mathsf{ord}}$.

That is, we show that though solving CAC is possible in any Turing ideal which contains solutions to all instances of $\mathsf{CAC}^{\mathsf{ord}}$ therein, it is not the case that every infinite poset P computes an ω-ordered poset $\widehat{P}$ with a solution $\widehat{X}$ such that $X \oplus P$ computes a solution of P. This is proven by analyzing the complexity of solutions to computable instances of $\mathsf{CAC}^{\mathsf{ord}}$. Herrmann [10] constructed an infinite computable poset which contains no infinite Δ^0_2 chain or antichain. We show next that any infinite ω-ordered poset which lacks an infinite computable chain must contain an infinite Δ^0_2 antichain.

Lemma 3.1.7. *If $(P, \leq_P)$ is an infinite computable ω-ordered poset and $(P, \leq_P)$ contains no infinite computable chain, then $(P, \leq_P)$ has an infinite Δ^0_2-definable antichain.*

Proof. Suppose $(P, \leq_P)$ is as hypothesized and assume without loss of generality that $P = \omega$. Notice the set X of elements of P with no strict $\leq_P$-successor is Π^0_1-definable:

$$X = \{x \in P : \forall y(x < y \to x \not\leq_P y)\}.$$

We claim every element of P has a successor in X. To see this, suppose not and fix $z \in P$ such that if $y \geq_P z$, then $y \notin X$. We computably construct an infinite chain above z. Indeed, there is a least n such that $n \geq_P z$ and $n \notin X$. Continuing, there is a least $m \geq_P n$ with $m \notin X$ and iterating this process yields an infinite computable chain in P, a contradiction.

We construct the infinite Δ^0_2-definable antichain by finding a suitable subset $Y \subseteq X$ computable in X. To begin, let x_0 be the least element in X. Assume by induction that x_n has been found. Let x_{n+1} be the least element of X such that $x_{n+1} \geq_p x_n + 1$.

Notice $x_{n+1} \mid_P x_n$ because x_n has no strict $\leq_P$ successor, and $x_{n+1} \geq x_n + 1 > x_n$. To conclude the construction, let $Y = \{x_n : n \in \omega\}$.

We claim Y is the desired Δ^0_2-definable antichain. By construction Y is infinite with $Y \leq_T X$. Since X is Δ^0_2-definable, Y must be as well. Since $Y \subseteq X$, no element of Y has a strict $\leq_P$-successor. In particular, there can be no pair $x_m, x_n \in Y$ such that $x_m \leq_P x_n$. Hence Y is an antichain, and the proof is complete. $\square$

Theorem 3.1.8. $\mathsf{CAC} \not\leq_{\mathsf{c}} \mathsf{CAC}^{\mathsf{ord}}$.

Proof. Let $(H, \leq_H)$ be Hermann's poset constructed in Theorem 3.1 of [10]. Then $(H, \leq_H)$ is infinite and contains no Δ^0_2-definable solution. Suppose $(P, \leq_P)$ is an infinite ω-ordered poset computable from $(H, \leq_H)$. If $(P, \leq_P)$ has a computable solution X, then X cannot compute a solution of $(H, \leq_H)$. If $(P, \leq_P)$ has no computable solution, then it must have a Δ^0_2 solution Y. Again, Y cannot compute a solution to $(H, \leq_H)$. $\square$

The key to finding Δ^0_2 solutions of $\mathsf{CAC}^{\mathsf{ord}}$ is noting that for any x in an ω-ordered poset $(P, \leq_P)$, if x has no strict successors then it cannot be in an infinite chain. Indeed, each x in P has at most x many strict predecessors. Thus the maximum elements in each maximal finite chain form an antichain. Here by "maximal finite chain" we mean one that is not contained in any larger finite chain.

This is not the case in general posets since any element may be a part of many infinite chains while still having no, or only finitely many, successors. This is exactly the nature of Herrmann's poset $(H, \leq_H)$. Its construction relies on repeatedly nesting new finite chains within previously constructed ones.

This observation motivates investigating the relationship between general stable posets and ω-ordered stable posets. Indeed, each element in an ω-ordered poset is either below or incomparable with infinitely many other elements. Thus each ω-ordered poset is very nearly a stable poset of the small type, but some element may be both comparable and incomparable with infinitely many elements. Note if $(P, \leq_P)$ is both stable and ω-ordered, then it is of the small type.

We first note that the proof of Theorem 3.1.5 goes through with CAC and $\mathsf{CAC}^{\mathsf{ord}}$ replaced by SCAC and $\mathsf{SCAC}^{\mathsf{ord}}$ respectively.

Corollary 3.1.9. *Over* RCA_0*, the principles* SCAC *and* $\mathsf{SCAC}^{\mathsf{ord}}$ *are equivalent.*

We also note that $\mathsf{CAC}^{\mathsf{ord}}$ strictly implies $\mathsf{SCAC}^{\mathsf{ord}}$ over RCA_0. This follows immediately from the work of Hirschfeldt and Shore [13]: in particular, Corollary 3.6 and Proposition 3.8 of [13] prove that CAC strictly implies SCAC over RCA_0.

Now, in contrast to CAC and $\mathsf{CAC}^{\mathsf{ord}}$, SCAC is computably equivalent to $\mathsf{SCAC}^{\mathsf{ord}}$. We show this next in Theorem 3.1.10. To do this, we take a given stable poset $(P, \leq_P)$ and computably refine it to a stable ω-ordered poset $(Q, \leq_Q)$ such that Q contains an infinite chain or infinite antichain X which, together with P, computes an infinite chain or infinite antichain in the original poset. The fundamental difference from Theorem 3.1.8 is that when refining $(P, \leq_P)$ to an ω-ordered poset, we can be sure that we will miss at most finitely many comparabilities for each element. For instance, if x is small in P, we know there are only finitely many $y > x$ such that $y \leq_P x$. If P is of the large type, we may simply 'flip the order' and again miss only finitely many comparabilities. In this way, we will be able to effectively thin any solution to the computed $\mathsf{SCAC}^{\mathsf{ord}}$ instance to a solution of $(P, \leq_P)$.

Theorem 3.1.10. $\mathsf{SCAC} \equiv_{\mathsf{c}} \mathsf{SCAC}^{\mathsf{ord}}$.

Proof. It is immediate that $\mathsf{SCAC}^{\mathsf{ord}} \leq_c \mathsf{SCAC}$, as every instance of $\mathsf{SCAC}^{\mathsf{ord}}$ is an instance of SCAC. To see that $\mathsf{SCAC} \leq_c \mathsf{SCAC}^{\mathsf{ord}}$, we fix an instance $(P, \leq_P)$ of SCAC

and assume without loss of generality that $P = \omega$. To proceed, we consider separately the case in which $(P, \leq_P)$ is of the small type and the case in which $(P, \leq_P)$ is of the large type. While the two cases will be entirely symmetric, the distinction will be necessary in the following construction.

Assume $(P, \leq_P)$ is a stable poset of the small type. Define $(Q, \leq_Q)$ so that $\leq_Q$ is the suborder of $\leq_P$ respecting the natural order on ω:

$$Q = P \text{ and } \leq_Q = \{(m, n) : m \leq n \wedge m \leq_P n\}.$$

Thus $(Q, \leq_Q) \leq_T (P, \leq_P)$ is an instance of $\mathsf{SCAC}^{\mathsf{ord}}$. Apply $\mathsf{SCAC}^{\mathsf{ord}}$ to obtain an infinite set X which is a solution of $(Q, \leq_Q)$.

As above, if X is a chain in Q, then it is chain in P; if instead X is an antichain, it may not be a solution of P. So we $(X \oplus P)$-computably thin X so that it is a solution in either case.

Define the predicate $R(m, n)$ to be

$$m \leq_Q n \leftrightarrow m \leq_P n.$$

So $R(m, n)$ holds if Q preserves the correct relationship between m and n and $R(m, n)$ fails only if $m \mid_Q n$ but $m \nmid_P n$. With $X = \{x_0 < x_1 < x_2 < \cdots\}$, define

$$X_0 = \{x_i \in X : R(x_0, x_i)\}.$$

Note $X_0 \subseteq X$ is infinite because $(P, \leq_P)$ is stable. The stability of $(P, \leq_P)$ guarantees that $(Q, \leq_Q)$ misses at most finitely many comparabilities for each element of P.

Assume we have built $X_n = \{x_0^n < x_1^n < x_2^n < \cdots\}$ where x_m^n is the mth element of X_n in increasing order. Construct $X_{n+1} \subseteq X_n$ as

$$X_{n+1} = \{x_i^n : i \leq n+1\} \cup \{x_i^n : i > n+1 \wedge R(x_{n+1}^n, x_i)\}.$$

Again, X_{n+1} is infinite because $(P, \leq_P)$ is stable. Let $Y = \lim_n X_n = \bigcap_{n \in \omega} X_n$. We claim that Y is infinite, Y is a solution of $(P, \leq_P)$, and $Y \leq_T X \oplus P$.

To see that Y is infinite, notice that for all n, $\left|\bigcap_{i \leq n} X_i\right| \geq n+1$, because in particular

$$\{x_0 < x_1^0 < x_2^1 < \cdots < x_{n+1}^n\} \subseteq \bigcap_{i=0}^{n} X_i.$$

To see that Y is a solution of $(P, \leq_P)$, note that if X was a chain in $(Q, \leq_Q)$, then $R(x, y)$ for all $x < y$ in X, and so $Y = X$. As $(x < y \wedge x \leq_Q y) \to x \leq_P y$, we conclude Y is a chain in $(P, \leq_P)$. If X was an antichain, then all $x < y$ in X are such that $x \mid_Q y$. By construction all $x < y$ in Y satisfy $R(x, y)$, so $x \mid_P y$. Thus Y is an antichain in $(P, \leq_P)$. Either way, Y is a solution of $(P, \leq_P)$.

Finally, to see that $Y \leq_T X \oplus P$, note $n \in Y$ if and only if $n \in X_n$, since at the nth stage of the construction, all elements less than or equal to n have been removed from X if they ever will be. Because each $X_n \leq_T X \oplus P$, we conclude $Y \leq_T X \oplus P$.

This completes the proof in the first case. The second case is symmetric: when $(P, \leq_P)$ is of the large type, define $(Q, \leq_Q)$ by

$$Q = P \text{ and } \leq_Q = \{(m, n) : m \leq n \wedge n \leq_P m\},$$

let $R(m, n)$ be the predicate

$$m \leq_Q n \leftrightarrow n \leq_P m,$$

and repeat the construction. □

Here we very nearly have a uniform construction. The single bit of non-uniform information used in this proof was whether or not the stable poset $(P, \leq_P)$ is of the small or large type. With this data given, the above construction would uniformly find solutions to instances of SCAC using $\mathsf{SCAC}^{\mathsf{ord}}$ in an effective manner. This motivates the definition of the following principles.

Statement 3.1.11. Let $(P, \leq_P)$ be an infinite stable poset. We define three principles $\mathsf{SCAC}^{\mathsf{small}}$, $\mathsf{SCAC}^{\mathsf{large}}$, $\mathsf{SCAC}^{\mathsf{type}}$ as follows:

$\mathsf{SCAC}^{\mathsf{small}}$: If $(P, \leq_P)$ is of the small type, then P contains an infinite chain or an infinite antichain.

$\mathsf{SCAC}^{\mathsf{large}}$: If $(P, \leq_P)$ is of the large type, then P contains an infinite chain or an infinite antichain.

$\mathsf{SCAC}^{\mathsf{type}}$: If $(P, \leq_P, T)$ is a triple with $T \in \{S, L\}$ such that $T = S$ (or L) implies $(P, \leq_P)$ is of the small (or large) type, then P contains an infinite chain or an infinite antichain.

Corollary 3.1.12. $\mathsf{SCAC}^{\mathsf{ord}} \equiv_{\mathsf{W}} \mathsf{SCAC}^{\mathsf{small}} \equiv_{\mathsf{W}} \mathsf{SCAC}^{\mathsf{large}} \equiv_{\mathsf{W}} \mathsf{SCAC}^{\mathsf{type}}$

Proof. In each case, apply the construction from Theorem 3.1.10 according to whether the given stable poset is of the small or large type. As this data is explicit in each instance, it can be hard-coded into the forward functional Φ for each reduction. □

Note that the proof of Theorem 3.1.10 requires the original instance of SCAC in the computation of its solution, so we do not obtain strong Weihrauch equivalences in Corollary 3.1.12. Indeed, Corollaries 3.2.6 and 3.3.9 below yield respectively that $\mathsf{SCAC} \not\leq_{\mathsf{W}} \mathsf{SCAC}^{\mathsf{ord}}$ and $\mathsf{SCAC} \not\leq_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{ord}}$. Together these imply the original instance is necessary in Corollary 3.1.12. By 'flipping the poset' we may obtain an analogue of Corollary 3.1.12 for strong computable reducibility.

Theorem 3.1.13. $\mathsf{SCAC} \equiv_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{small}}$.

Proof. Trivially, we obtain $\mathsf{SCAC}^{\mathsf{small}} \leq_{\mathsf{sc}} \mathsf{SCAC}$. To show $\mathsf{SCAC} \leq_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{small}}$, suppose $(P, \leq_P)$ is an instance of SCAC. If $(P, \leq_P)$ is of the small type, we may witness this reduction with the identity functionals. If $(P, \leq_P)$ is of the large type, we compute from $(P, \leq_P)$ the *dual* partial order $(P, \leq'_P)$, i.e., $\leq'_P$ is defined by

$$x \leq'_P y \text{ if and only if } y \leq_P x.$$

We claim $(P, \leq'_P)$ is a stable poset of the small type. Indeed, if x is $\leq_P$-isolated, then clearly x is $\leq'_P$-isolated. If x is $\leq_P$-large, then there is a t such that if $y > t$, then $x \geq_P y$. This implies that for each $y > t$, $x \leq'_P y$. So x is $\leq'_P$-small. This verifies the claim.

Apply $\mathsf{SCAC}^{\mathsf{small}}$ to $(P, \leq'_P)$ to obtain an infinite set $X \subseteq P$ which is either a chain or antichain in $\leq'_P$. If X is an antichain in $\leq'_P$, it is an antichain in $\leq_P$. If X is a chain in $\leq'_P$, it a chain $\leq_P$ in the opposite direction. In either case, X is identically a solution of $(P, \leq_P)$. $\square$

Corollary 3.1.14. $\mathsf{SCAC} \equiv_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{small}} \equiv_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{large}} \equiv_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{type}}$.

Proof. It remains to show that $\mathsf{SCAC} \leq_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{large}}$ and $\mathsf{SCAC} \leq_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{type}}$. For $\mathsf{SCAC}^{\mathsf{type}}$, take a forward functional which appends S to $(P, \leq_P)$ if it is of the small type, else take one that appends L; the backward functional is the identity. For $\mathsf{SCAC}^{\mathsf{large}}$, proceed symmetrically to the proof of Theorem 3.1.13. □

Again, as we show $\mathsf{SCAC} \not\leq_{\mathsf{W}} \mathsf{SCAC}^{\mathsf{ord}}$ and $\mathsf{SCAC} \not\leq_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{ord}}$ below, we have that the proof of Theorem 3.1.13 cannot be made uniform and hence Corollary 3.1.14 is optimal.

3.2 Weihrauch reducibility and $\mathsf{SCAC}^{\mathsf{ord}}$

In this section we prove that Theorem 3.1.10 cannot be improved with respect to Weirauch reductions. That is, we show $\mathsf{SCAC} \not\leq_{\mathsf{W}} \mathsf{SCAC}^{\mathsf{ord}}$. In section 3.3 we show the same with respect to strong computable reductions, namely $\mathsf{SCAC} \not\leq_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{ord}}$. In each case, we will need to build an instance of SCAC witnessing the non-reduction. This is done via the forcing notion $\mathbb{P}$. While the main proof in this section will be a finite extension argument organized by $\mathbb{P}$, we will require a more general forcing argument in section 3.3.

We assume familiarity with forcing in arithmetic (see section 2.3 of [11] for a basic overview). The conditions in this case will be finite posets π on an initial segments of ω with an *assignment function* a that locks each element in π to a stable limit behavior. We adapt the forcing notion $\mathbb{P}$ used in section 4 of Astor, Dzhafarov, Solomon, and Suggs [1].

To begin, let FinPO be the set of all finite partial orders on initial segments of ω. For each $\pi \in \text{FinPO}$, let $\leq_\pi$ denote the order relation in π and $|\pi|$ be the greatest n such that π orders $\omega \upharpoonright n$. So $\pi = (\omega \upharpoonright |\pi|, \leq_\pi) = (|\pi|, \leq_\pi)$. We say a poset $(P, \leq_P)$ *extends* and is *compatible* with π^p if $\omega \upharpoonright |\pi| \subseteq P$, and for all $x, y < |\pi|$ we have $x \leq_P y$

if and only if $x \leq_\pi y$. Thus if $\rho \in \mathrm{FinPO}$ extends π then $|\rho| \geq |\pi|$. We canonically code each finite set $\pi \in \mathrm{FinPO}$ so the map $\pi \mapsto |\pi|$ is effective.

Definition 3.2.1. Let $\mathbb{P}$ be the following notion of forcing. A *condition* is a pair $p = (\pi^p, a^p)$ as follows:

- $\pi^p \in \mathrm{FinPO}$ where FinPO is the set of all partial orders on initial segments of ω.
- a^p is a map $|\pi^p| \to \{S, L, I\} \times (\omega \restriction |\pi^p| + 1)$ such that: either for all $n < |\pi^p|, a^p(n) \in \{S, I\} \times (\omega \restriction |\pi^p| + 1)$, or for all $n < |\pi^p|, a^p(n) \in \{L, I\} \times (\omega \restriction |\pi^p| + 1)$;
- if $a^p(x) = (S, t)$ and $y \leq_{\pi^p} x$, then $y < t$ and $a^p(y) = (S, u)$ for some u;
- if $a^p(x) = (L, t)$ and $x \leq_{\pi^p} y$, then $y < t$ and $a^p(y) = (L, u)$ for some u;
- if $a^p(x) = (S, t)$ or $a^p(x) = (L, t)$ and $x \,|_{\pi^p}\, y$, then $y < t$; and
- if $a^p(x) = (I, t)$ and $x \leq_{\pi^p} y$ or $y \leq_{\pi^p} x$, then $y < t$.

A condition q *extends* p, written $q \leq_{\mathbb{P}} p$, if π^q extends π^p and $a^q \subseteq a^p$.

Intuitively, the map a^p *assigns* to each $x < |\pi^p|$ a limit behavior and a *stabilization point* t so that any other element $y \geq t$ in π^p relates to x in the correct way (e.g., is above x if x is small). Clearly any filter $\mathfrak{F} \subseteq \mathbb{P}$ gives rise to a stable poset given by $\bigcup_{p \in \mathfrak{F}} \pi^p$. We use $G = (\omega, \leq_G)$ to denote this poset and also use G as a name for the generic poset in $\mathbb{P}$ forcing language.

We prove $\mathsf{SCAC} \not\leq_{\mathsf{W}} \mathsf{SCAC}^{\mathsf{ord}}$ by contradiction. So, we assume there is a fixed pair of functionals, Φ and Ψ, which witness that $\mathsf{SCAC} \leq_{\mathsf{W}} \mathsf{SCAC}^{\mathsf{ord}}$. We then construct a computable stable poset $G = (\omega, \leq_G)$, such that the stable ω-ordered poset Φ^G

contains an infinite chain or antichain X, for which Ψ^X is not a solution of G. We will achieve this in two ways. The first is to ensure G has no computable solutions, so that if there is a computable solution X of Φ^G, the set defined by Ψ^X will be computable, and thus not a solution of G. The second is to directly diagonalize a solution X in Φ^G by ensuring either that it contains both comparable and isolated elements in G, or that it contains both incomparable and small elements in G.

Both goals will be met by a finite extension argument. The following lemma ensures that from any finite poset constructed, we may computably obtain a compatible infinite poset G without computable solutions.

Lemma 3.2.2. *For any condition $p \in \mathbb{P}$ there is an infinite computable stable poset G compatible with p such that G has no infinite computable chain or infinite computable antichain.*

Proof. Fix $p \in \mathbb{P}$. Let $\widehat{G} = (J, \leq_{\widehat{G}})$ be a computable stable poset with no infinite computable chain or infinite computable antichain. Without loss of generality, assume $J = \omega - |\pi^p|$. We define $G = (\omega, \leq_G)$ as follows:

- $\leq_G^{|\pi^p|} = \leq_{\pi^p}$;
- $\leq_G^J = \leq_{\widehat{G}}$;
- and for each $x < |\pi^p|$:
  - if $a^p(x) = (I, t)$, set $x \mid_G y$ for all $y \in J$;
  - if $a^p(x) = (S, t)$, set $x <_G y$ for all $y \in J$;
  - if $a^p(x) = (L, t)$, set $y <_G x$ for all $y \in J$.

Clearly G is a computable stable poset compatible with p. Let X be an infinite chain or infinite antichain in G. Then X is not computable. Indeed, if X is computable, then $X - |\pi^p|$ is an infinite computable chain or infinite computable antichain in $\widehat{G}$, a contradiction. $\square$

To find a solution in Φ^G which we can diagonalize, we apply a Seetapun-style construction. By finitely extending a given condition p in a uniformly computable manner to a condition q with $|\pi^q|$ sufficiently large, we obtain witnesses x, y from finite sets F in Φ^{π^q} such that $\Psi^F(x) \downarrow = \Psi^F(y) \downarrow = 1$. The advantage we will have is that no matter if x and y are made comparable or incomparable by π^q, we may require a^q to assign limit behavior to these elements which diagonalize any solution of Φ^G extending F. The Seetapun configuration will be achieved when sufficiently many finite chains $F_0, \ldots, F_n$ have been found and diagonalized in Φ^G to ensure an antichain E made of elements from each chain F_i can be diagonalized as well. The key combinatorial lemma that will allow us to simultaneously prevent $F_0, \ldots, F_n$ and E from extending to a solution is presented next. The idea is that as the limit behavior of the elements in π^q do not affect the local structure of π^q, we may find a suitable condition which both guarantees the witnesses from $E, F_0, \ldots, F_n$ and simultaneously diagonalizes them via limit behavior in G (e.g., if $x \leq_G y$, we can make x small and y isolated).

We call two conditions $p, q \in P$ *parallel* if $\pi^p = \pi^q$. Notice if p and q are parallel conditions, then for any functional Φ, we have $\Phi^{\pi^p} = \Phi^{\pi^q}$. Thus if we need to change the limit assignments in some condition p, we can computably do so while maintaining the structure of Φ^{π^p}. This will be ensured so long as the condition we move to is parallel to p. Since there are only finitely many conditions parallel to each p, we may effectively check all of them at some stage in a computable construction. In the proof of Theorem 3.2.5, we will need a systematic way to move from a condition p constructing a stable poset of the small type to a parallel condition q constructing a poset of the large type. We prove we may "change our mind" about the sort of stable poset being built at any point in the construction.

Lemma 3.2.3. *If $p \in \mathbb{P}$ is such that the range of a^p is contained in $\{S, I\} \times \omega \restriction |\pi^p|+1$, then there is a parallel condition $q \in \mathbb{P}$ such that for all $x < |\pi^p|$*

$$(a^p(x) = (S,t) \to a^q(x) = (I, |\pi^p|)) \wedge (a^p(x) = (I,t) \to a^q(x) = (L,t)).$$

Proof. Fix p and let q be the pair (π^p, a^q) with a^q defined as follows: for each $x < |\pi^p|$ if $a^p(x) = (S,t)$ for some t, then $a^q(x) = (I, |\pi^p|)$, and if $a^p(x) = (I,t)$, then $a^q(x) = (L,t)$. We show that $q \in P$. Obviously $\pi^p \in \mathrm{FinPO}$ so it remains to show a^q satisfies the required properties.

Fix an arbitrary $x < |\pi^p|$. Note $a^q(x) \neq (S,t)$ for any t. Suppose $a^q(x) = (I,t)$ for some t. Then $t = |\pi^p|$ by definition. So if $y \leq_{\pi^p} x$ or $x \leq_{\pi^p} y$, we have $y < |\pi^p|$ because $y \in \pi^p$, satisfying the required property.

Suppose $a^q(x) = (L,t)$ for some t. Then $a^p(x) = (I,t)$ by definition. So if $x \leq_{\pi^p} y$ then $y < t$. We claim $a^p(y) = (I,u)$ for some u, which implies $a^q(y) = (L,u)$, satisfying the needed property. To see this, suppose not: by hypothesis, $a^p(y) = (S,u)$ for some u. Since $x \leq_{\pi^p} y$, we have $x < u$ and $a^p(x) = (S,v)$ for some v. But $a^p(x) = (I,t)$, a contradiction. $\square$

This ability to "change our mind" at any point, and construct a stable poset of the large type, is the core combinatorial advantage of SCAC that we will exploit over

$\mathsf{SCAC}^{\mathsf{ord}}$. Any instance of $\mathsf{SCAC}^{\mathsf{ord}}$ must be of the small type, while we may freely decide at any point in the construction which type of stable poset we produce.

Before proving the desired result, we require one additional lemma. The construction will produce G uniformly computably in the indices for the given pair of functionals Φ and Ψ. The upshot of this is that we will obtain $\mathsf{SCAC} \not\leq_{\mathsf{W}} \mathsf{SCAC}^{\mathsf{ord}}$ without having to join the initial segments of G with the finite sets F we are finding in Φ^G. Lemma 3.2.4 guarantees this is possible.

Lemma 3.2.4. *Given two problems* P *and* Q*, if for each pair of functionals* Φ *and* Ψ*, there is a computable instance of* P *uniformly computable in (the indices of) these functionals which witnesses* $\mathsf{P} \not\leq_{\mathsf{sW}} \mathsf{Q}$ *(via* Φ *and* Ψ*), then* $\mathsf{P} \not\leq_{\mathsf{W}} \mathsf{Q}$*.*

Proof. Suppose P and Q are as hypothesized and $f(i,j)$ is a computable function which outputs (the index of) the instance X of P witnessing $\mathsf{P} \not\leq_{\mathsf{sW}} \mathsf{Q}$ via the functionals Φ_i and Φ_j. For sake of contradiction assume that Φ_m and Φ_n witness $\mathsf{P} \leq_{\mathsf{W}} \mathsf{Q}$. Define a computable function $g(k)$ such that with m and n fixed we have

$$\Phi_{g(k)}^{Y} = \Phi_n^{\Phi_{f(m,k)} \oplus Y}.$$

Via the relativized recursion theorem, find a fixed point k such that for all Y

$$\Phi_{g(k)}^{Y} = \Phi_k^{Y}.$$

We claim that Φ_m and Φ_k contradict $f(m,k)$, i.e., Φ_m and Φ_k witness $\mathsf{P} \leq_{\mathsf{sW}} \mathsf{Q}$ with $X = \Phi_{f(m,k)}$. To see this, note that f guarantees a solution $\widehat{Y}$ to $\Phi_m^{\Phi_{f(m,k)}} = \Phi_m^X$ such that $\Phi_k^{\widehat{Y}}$ is *not* a solution to $X = \Phi_{f(m,k)}$. But by construction

$$\Phi_k^{\widehat{Y}} = \Phi_{g(k)}^{\widehat{Y}} = \Phi_n^{\Phi_{f(m,k)} \oplus \widehat{Y}} = \Phi_n^{X \oplus \widehat{Y}}.$$

Since we assumed Φ_n witnesses, along with Φ_m, that $\mathsf{P} \leq_{\mathsf{W}} \mathsf{Q}$, we have that $\Phi_n^{X \oplus \widehat{Y}} = \Phi_k^{\widehat{Y}}$ *is* a solution to $X = \Phi_{f(m,k)}$, a contradiction. Whence, we conclude that $\mathsf{P} \not\leq_{\mathsf{W}} \mathsf{Q}$. $\square$

We are now ready to prove the desired non-reduction. To summarize the approach, we will suppose by way of contradiction that Φ and Ψ witness $\mathsf{SCAC} \leq_{\mathsf{sW}} \mathsf{SCAC}^{\mathsf{ord}}$. We will then build a stable poset $G = (\omega, \leq_G)$ uniformly computably in (the indices of) Φ and Ψ by computably finding finite extensions of conditions in $\mathbb{P}$. The construction will conclude with Lemma 3.2.2 to ensure G has no computable solution. If necessary, we will apply Lemma 3.2.3 at one point in the construction to make G a stable poset of the large type. Otherwise, G will be of the small type. This will be required depending on the interaction of witness from finite sets $E, F_0, \ldots, F_n$ in Φ^{π^p} where π^p is an initial segment of G. Finally, Lemma 3.2.4 will ensure this yields that $\mathsf{SCAC} \not\leq_{\mathsf{W}} \mathsf{SCAC}^{\mathsf{ord}}$.

Theorem 3.2.5. $\mathsf{SCAC} \not\leq_{\mathsf{sW}} \mathsf{SCAC}^{\mathsf{ord}}$.

Proof. By way of contradiction, assume the functionals Φ and Ψ witness $\mathsf{SCAC} \leq_{\mathsf{sW}} \mathsf{SCAC}^{\mathsf{ord}}$. So if $P = (\omega, \leq_P)$ is a stable poset, Φ^P is an ω-ordered stable poset and any infinite chain or infinite antichain X in Φ^P produces an infinite chain or infinite antichain Ψ^X in P. We construct a computable stable poset $G = (\omega, \leq_G)$ which diagonalizes this pair of functionals. That is, we construct G in an effective manner which ensures Φ^G contains either an infinite computable chain or an infinite antichain X such that Ψ^X is not a solution to G. For the latter case, we will ensure that Ψ^X contains a pair $\{a, b\}$ such that if a and b are comparable in G then one of them is isolated, and if a and b are incomparable then one of a or b is either small or large. For the former case, we will conclude our construction with an application of Lemma 3.2.2 to guarantee G has no infinite computable chain or infinite computable antichain. As this can be done at any condition $p \in \mathbb{P}$, we assume any poset G taken below to extend the current condition has this property.

Begin by constructing a condition p for which π^p is an initial segment of the natural order on ω and a^p assigns S or I to each element in $|\pi^p|$. Make $|\pi^p|$ sufficiently large to reveal a finite chain F_0 in Φ^{π^p} such that there exists a pair $x_0, y_0 < |\pi^p|$ with $\Psi^{F_0}(x_0) \downarrow= \Psi^{F_0}(y_0) \downarrow= 1$. Such a p and F_0 must exist. If not, fix p for which this fails. Apply Lemma 3.2.2 to obtain an infinite computable stable poset G compatible with p that has no infinite computable chain or infinite computable antichain. If Φ^G has an infinite chain X, the set defined by Ψ^X is finite. If Φ^G has no infinite chain then cofinitely many elements in Φ^G are isolated. Thus Φ^G has an infinite computable antichain X and so Ψ^X is not a solution of G. In either case, Φ and Ψ fail to witness $\mathsf{SCAC} \leq_{\mathsf{sW}} \mathsf{SCAC}^{\mathsf{ord}}$, a contradiction.

The only property of p needed to find F_0 is that $|\pi^p|$ is sufficiently large. So we can further require that $a^p(x_0) = (S, |\pi^p|)$ and $a^p(y_0) = (I, |\pi^p|)$ if $x_0 <_{\pi^p} y_0$, or $a^p(x_0) = (I, |\pi^p|)$ and $a^p(y_0) = (S, |\pi^p|)$ otherwise. Note $\max F_0$ will be isolated in Φ^G. To see this, suppose otherwise: then there is a condition $q \leq_{\mathbb{P}} p$ such that the resulting G has $\max F_0$ is small in Φ^G. From above we know Φ^G must have an infinite chain. In particular then, Φ^G must have an infinite chain $F_0 \cup X$ with $F_0 < X$. By construction, the set $\Psi^{F_0 \cup X}$ has both a small element and an isolated element and so is not a solution of G. This contradicts that Φ and Ψ witness the reduction.

Assume we are at condition q_{n-1} and have found finite sets $F_0, \ldots, F_{n-1}$ such that for each $i \leq n-1$

- F_i is a chain in $\Phi^{\pi^{q_{n-1}}}$ and q_{n-1} ensures $\max F_i$ will be isolated in Φ^G;
- $\min F_i \mid_{\Phi^{\pi^{q_{n-1}}}} \max F_j$ for all $j < i$;
- there is a pair x_i, y_i such that $\min\{x_i, y_i\} > \max\{x_j, y_j : j < i\}$ and $\Psi^{F_i}(x_i) \downarrow = \Psi^{F_i}(y_i) \downarrow = 1$; and
- $a^{q_{n-1}}(x_i) = (S, |\pi^{q_{n-1}}|)$ and $a^{q_{n-1}}(y_i) = (I, |\pi^{q_{n-1}}|)$ if $x_i <_{\pi^{q_{n-1}}} y_i$, or $a^{q_{n-1}}(x_i) = (I, |\pi^{q_{n-1}}|)$ and $a^{q_{n-1}}(y_i) = (S, |\pi^{q_{n-1}}|)$

Find a condition $q_n \leq_{\mathbb{P}} q_{n-1}$ for which there is a chain F_n in $\Phi^{\pi^{q_n}}$ such that both $\Psi^{F_n}(x_n) \downarrow = \Psi^{F_n}(y_n) \downarrow = 1$ for some pair $x_n, y_n < |\pi^{q_n}|$ with $\min\{x_n, y_n\} > \max\{x_i, y_i : i < n\}$, and $\min F_n \mid_{\Phi^{\pi^{q_n}}} \max F_i$ for each $i < n$. As before, finding F_n depends only on $|\pi^{q_n}|$ and not a^{q_n}, so by moving to a parallel condition if necessary, we can ensure q_n additionally satisfies that $a^{q_n}(x_n) = (S, |\pi^{q_n}|)$ and $a^{q_n}(y_n) = (I, |\pi^{q_n}|)$ if $x_n <_{\pi^{q_n}} y_n$, or $a^{q_n}(x_n) = (I, |\pi^{q_n}|)$ and $a^{q_n}(y_n) = (S, |\pi^{q_n}|)$ otherwise. Note the only worry is if the assigned limit behavior of x_n and y_n causes conflict with a previous pair x_i, y_i. To avoid this, take x_n and y_n sufficiently large so that they also surpass the stabilization points of each x_i and y_i.

Again such a q_n and F_n must exist. If not, we reach a contradiction similar to before: apply Lemma 3.2.2 with q_{n-1} to obtain the stable poset G. Note $\max F_i$ is isolated in Φ^G for each $i < n$ and Φ^G must contain an infinite chain. Thus, there is some infinite chain X with $\min X \mid_{\Phi^G} \max F_i$ for all $i < n$. Furthermore, for every initial segment F of X, there is no sufficiently large pair x_n, y_n which enters the set defined by Ψ^F. Thus Ψ^X will define a finite set, and we again contradict that Φ and Ψ witness $\mathsf{SCAC} \leq_{\mathsf{sW}} \mathsf{SCAC}^{\mathsf{ord}}$.

Let $E = \{\max F_i : i \leq n\}$. Note E is an antichain in $\Phi^{\pi^{q_n}}$ and every element of E

must be isolated in Φ^G. Without loss of generality, assume n and $|\pi^{q_n}|$ are sufficiently large so that there is a pair $x, y < |\pi^{q_n}|$ with $\Psi^E(x) \downarrow = \Psi^E(y) \downarrow = 1$. If no such n exists, then we can continue the sequence $F_0, F_1, \dots F_n$ *ad infinitum* to extend E to an infinite antichain X in Φ^G for which Ψ^X is finite, a contradiction.

The final step in our construction is to simultaneously diagonalize E and each set $F_0, \dots, F_n$. Let $D = \{x_i, y_i : i \leq n\}$ be the set of witnesses for $F_0, \dots, F_n$. Here we cannot guarantee that x and y are outside of D or that they are beyond the stabilization points of each element in D. Thus we seek a condition r parallel to q_n such that a^r diagonalizes each of $E, F_0, \dots, F_n$. For E, we will ensure a^r makes at least one of x or y isolated if they are comparable in π^{q_n}, or at least one of x or y small otherwise. To ensure the elements of E are isolated, we need to maintain the diagonalization of the chains $F_0, \dots, F_n$. If there is a chain F_i whose diagonalization we cannot maintain, we re-diagonalize the pair x_i, y_i similar to x or y depending on the comparability of x_i and y_i in π^{q_n}.

There are three cases: either $x <_{\pi^{q_n}} y$, $y <_{\pi^{q_n}} x$, or $x \mid_{\pi^{q_n}} y$. The first two are symmetric so we assume without loss of generality that $x <_{\pi^{q_n}} y$ or $x \mid_{\pi^{q_n}} y$.

> **Case 1:** $x <_{\pi^{q_n}} y$. If possible, move to a condition r parallel to q_n such that $a^r(y) = (I, |\pi^{q_n}|)$ and for each $z \in D$ we have $a^r(z) = a^{q_n}(z)$. If not, then since y cannot be made isolated while respecting the current limit behaviors assigned to the elements in D, there must be some $s \in D$ such that $y <_{\pi^{q_n}} s$ and $a^{q_n}(s) = (S, t_1)$ for some t_1. We claim in this case x can be made small, i.e., there is a parallel condition r' with $a^r(x) = (S, |\pi^{q_n}|)$ and $a^r(z) = a^{q_n}(z)$ for each $z \in D$. If not, then similarly there is some element $i \in D$ such that $i <_{\pi^{q_n}} x$ and $a^{q_n}(i) = (I, t_2)$ for some t_2. But

then by transitivity $i <_{\pi^{q_n}} s$, so $a^{q_n}(i) = (S, u)$ for some u, a contradiction. So there is a condition r' parallel to q_n, which makes both x and y small, since y cannot be isolated, and which agrees with q_n on the limit behavior of the elements in D. Apply Lemma 3.2.3 to r' to obtain the parallel condition r which makes every small element in π^{q_n} isolated and every isolated element in π^{q_n} large. Clearly, r diagonalizes E as a^r makes both x and y isolated. To see that r diagonalizes each of $F_0, \ldots, F_n$, fix F_i for some $i \leq n$. If $a^{q_n}(x_i) = (S, t)$ and $a^{q_n}(y_i) = (I, u)$ for some t and u then $x_i <_{\pi^{q_n}} y_i$. Since a^r makes x_i isolated and y_i large, Ψ^{F_i} still contains two comparable elements one of which is isolated. If on the other hand $a^{q_n}(x_i) = (I, t)$ and $a^{q_n}(y_i) = (S, u)$ then $y_i <_{\pi^{q_n}} x_i$ or $x_i \mid_{\pi^{q_n}} y_i$. As a^r makes x_i large and y_i isolated, Ψ^{F_i} either contains both comparable and isolated elements in the first case or both incomparable and large elements in the latter case. Thus r is the desired condition.

Case 2: $x \mid_{\pi^{q_n}} y$. Here we need to make either x or y small so suppose this is not possible while respecting the limit behaviors of the elements in D. Then there is a condition r' parallel to q_n such that $a^{r'}(x) = a^{r'}(y) = (I, |\pi^{q_n}|)$ and $a^{r'}(z) = a^{q_n}(z)$ for all $z \in D$. Apply Lemma 3.2.3 to r' to obtain a condition r in which every small element in π^{q_n} is made isolated and every isolated element in π^{q_n} is made large. As in Case 1, r is the desired condition. In particular, a^r makes x and y both large, so Ψ^E contains two incomparable large elements.

To conclude, note that the extension r was found computably. Indeed, the search for each F_i must succeed and we can computably extend any condition to conduct this search. For any condition p, we may bound the number of parallel conditions considered by fixing a maximum stabilization point, so setting the correct limit behaviors was also computable. Apply Lemma 3.2.2 to obtain an infinite computable stable poset G extending r with no infinite computable chain or infinite computable antichain. Hence Φ^G is an ω-ordered stable poset containing an infinite antichain. In particular Φ^G has an infinite antichain $E \cup X$ with $E < X$. By construction, $\Psi^{E \cup X}$ contains two elements x and y. If x and y are comparable in G, then one of these elements is isolated. If x and y are incomparable in G, then one of these elements is small if G is of the small type, or one is large if G is of the large type. Either way, $\Psi^{E \cup X}$ is not an infinite chain or infinite antichain in G. This contradicts the initial assumption that Φ and Ψ witness $\mathsf{SCAC} \leq_{\mathsf{sW}} \mathsf{SCAC}^{\mathsf{ord}}$ and the proof is complete. $\square$

Corollary 3.2.6. $\mathsf{SCAC} \not\leq_{\mathsf{W}} \mathsf{SCAC}^{\mathsf{ord}}$

In view of Corollary 3.1.12, we also obtain the following as mentioned above.

Corollary 3.2.7. *The principle* SCAC *is not Weihrauch equivalent to any of the following principles:* $\mathsf{SCAC}^{\mathsf{small}}$*,* $\mathsf{SCAC}^{\mathsf{large}}$*, and* $\mathsf{SCAC}^{\mathsf{type}}$*.*

3.3 Strong computable reducibility and $\mathsf{SCAC}^{\mathsf{ord}}$

We now turn our attention to strong computable reducibility. Specifically, we show that $\mathsf{SCAC} \not\leq_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{ord}}$. In the previous section, we have shown that for each fixed pair of functionals Φ and Ψ, we can computably build a sufficiently large finite poset π which ensures sufficiently large chains and antichains in Φ^π to diagonalize Ψ with. In a computable reduction, we are concerned with *any* stable ω-ordered poset computable from our instance G of SCAC. Thus, we must diagonalize Φ^G for every functional Φ

simultaneously. This means we can not reliably find fresh pairs (x, y) to diagonalize some collection of finite chains for a given forward functional. While trying to find a pair to diagonalize some finite chain in Φ_0^π say, we may repeatedly require that pair to diagonalize finite chains in other instances Φ_i^π.

To show SCAC is not strongly-computably reducible to $\mathsf{SCAC}^{\mathsf{ord}}$, we require a new approach. The idea is to exploit the local structure of an ω-ordered poset. In our instance $G = (\omega, \leq_G)$ of SCAC, we may freely set $x <_G y$ or $y <_G x$ regardless of the order of x and y in ω. So we seek to establish a situation in which the order of a pair (x, y) in G determines the order of some related pair (a, b) in Φ^G. We can then trap Φ^G into failing to be ω-ordered by forcing $b <_{\Phi^G} a$ when $a < b$. In this way, the core combinatorial difference in SCAC and $\mathsf{SCAC}^{\mathsf{ord}}$ is formally revealed. Of course, if there is a relatively simple solution in Φ^G from which we can avoid computations, we will do so.

Contrast this with the construction in section 3.2, which exploited the global structure of an ω-ordered poset. Namely, that any stable ω-ordered poset must be of the small type. Here, we will obtain that even $\mathsf{SCAC}^{\mathsf{small}} \not\leq_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{ord}}$.

While the basic idea is to force some $b > a$ to be below a in the order $\leq_{\Phi^G}$, creating this situation will require intricate machinery. As in the previous result, the stable poset $G = (\omega, \leq_G)$ will be constructed using the forcing notion $\mathbb{P}$ with $G = \bigcup_n \pi^{p_n}$ for a sequence $\langle p_n \rangle_{n\in\omega}$ of $\mathbb{P}$-conditions. However, in this construction, G will be far from computable. Instead we must take G sufficiently generic. This is because we will also construct a countable family of sets $\mathcal{D}$ each of which (when joined with G) will be unable to compute an infinite chain or infinite antichain in G. The next lemma shows that we may ensure G has this property.

Lemma 3.3.1. *There is an n such that for any set R, if G is the poset resulting from a $\Sigma^0_n(R)$-generic filter in $\mathbb{P}$, then G contains no $(G \oplus R)$-computable solution.*

Proof. Fix a set R and functional Γ. Let W be the set of $\mathbb{P}$ conditions that force one of the following:

- The set defined by $\Gamma^{G\oplus R}$ is finite.
- There are two elements $x, y \in \Gamma^{G\oplus R}$ such that x is isolated in G and y is either small or large in G.

Clearly W is uniformly arithmetic in R. So there is some n such that W is $\Sigma^0_n(R)$. It remains to show that W is dense in $\mathbb{P}$. To see this, let p be any condition in $\mathbb{P}$. Either there is a $q \leq_{\mathbb{P}} p$ forcing $\Gamma^{G\oplus R}$ to define a finite set, in which case $q \in W$, or p forces that $\Gamma^{G\oplus R}(x) \downarrow = 1$ for infinitely many x. In the latter case, there is a condition $r \leq_{\mathbb{P}} p$ and two numbers $x, y > |\pi^p|$ such that $\Gamma^{\pi^r \oplus R}(x) \downarrow = \Gamma^{\pi^r \oplus R}(y) \downarrow = 1$. As the limit behavior of x and y does not effect the local structure of π^r, we can further assume without loss of generality that $a^r(x) = (I, t_0)$ and $a^r(y) = (S, t_1)$ for some t_0 and t_1. Thus $r \in W$ and the proof is complete. $\square$

For each valid forward functional Φ that sends stable posets G to stable ω-ordered posets Φ^G, we define two infinite sets C_Φ and A_Φ that are respectively a chain and antichain in Φ^G. The goal will be to ensure at least one of these sets cannot compute a solution of G. That is, for some $X \in \{C_\Phi, A_\Phi\}$ we have that Ψ^X is not an infinite chain or antichain in G for any functional Ψ. If this proves impossible, we will obtain a suitable set R to add to $\mathcal{D}$ and in this way diagonalize Φ for any backward functional Ψ. One of these two approaches must succeed, or else we find ourselves able to show that some pair $a < b$ has $b <_{\Phi^G} a$.

The construction will handle one triple of functionals (Φ, Ψ_0, Ψ_1) at a time. The first, Φ, will be the forward functional giving us an stable ω-ordered poset Φ^G. The latter two will be treated as potential backwards functionals in a computable reduction involving the instance Φ^G. We will seek to ensure at least one of them fails to render a solution to our instance G. This will be done by either forcing $\Psi_0^{C_\Phi}$ or $\Psi_1^{A_\Phi}$ to be a set which cannot be a solution of G (e.g., because it is finite or contains both small and incomparable elements). We call accomplishing the first task "making progress on the chain" and accomplishing the second task "making progress on the antichain." It will suffice to make progress on only one of theses sets for each triple (Φ, Ψ_0, Ψ_1). This follows from Lachlan's Disjunction (Proposition 3.3.2) with $\mathcal{P}(X)$ being the predicate which says X is not an infinite chain or antichain in G. If we always make progress on the chain or make progress on the antichain, in the end, one of C_Φ or A_Φ will not compute a solution of G via any functional Ψ.

Proposition 3.3.2 (Lachlan's Disjunction)**.** *Given sets C and A, let $\mathcal{P}$ be a property of sets. If for any pair of functionals (Ψ_0, Ψ_1), we have $\mathcal{P}(\Psi_0^C)$ or $\mathcal{P}(\Psi_1^A)$ then for some $X \in \{C, A\}$ we have $\mathcal{P}(\Psi^X)$ for all functionals Ψ.*

Proof. The contrapositive is a tautology. Fix C and A. If there are functionals Ψ_0 and Ψ_1 such that $P(\Psi_0^C)$ does not hold and $P(\Psi_1^A)$ does not hold, then there is a pair of functionals, namely (Ψ_0, Ψ_1), such that both $P(\Psi_0^C)$ and $P(\Psi_1^A)$ fail to hold. □

The bulk of technical work will arise in finding appropriate finite extensions of initial segments of C_Φ and A_Φ. Call these initial segments C and A. To make progress on the chain, we will to need find a finite set F such that $\Psi_0^{C \cup F}$ presents a diagonalization opportunity against being solution to G. If this fails, we will attempt to make progress on the antichain and find a similar finite set F for which $\Psi_1^{A \cup F}$ can be prevented from forming a solution to G. We conduct these searches symmetrically, beginning first with the chain C, before repeating the search in an attempt to extend A. The key step will be to gain leverage on the global structure of Φ^G after each failed search. When both searches fail, we will gain total control on the limit behavior of a particular set of elements in Φ^G and become able to trap Φ^G from being both stable and ω-ordered as mentioned above.

To conduct each search, we rely on a technique known as *tree labeling*. This will be the tool by which we gain control over the limit behavior of elements in Φ^G. This framework was first introduced in [6], and subsequently applied in [7] and [21]. The rough idea is to collect all potential extensions of a Mathias condition (E, I) that present a diagonalization opportunity against some functional Δ. That is, we organize all finite sets $F \subseteq I$ which have $\Delta^{E\cup F}(w) \downarrow= 1$ for some sufficiently large w that has yet to be committed in the construction. The formal definition of this technique is given next.

Definition 3.3.3. For strings, $\alpha, \beta \in \omega^\omega$, we let $a^\#$ abbreviate $\alpha \upharpoonright |\alpha| - 1$ and $\alpha * \beta$ denote the concatenation of α and β. For any $x \in \omega$, we let $\alpha * x = \alpha * \langle x \rangle$. Thus $(\alpha * x)^\# = \alpha$. If $T \subseteq \omega^{<\omega}$ is a tree, then for any $\alpha \in T$, we call the set $R = \{x \in \omega : \alpha * x \in T\}$ the *row below* α.

Definition 3.3.4. The *extension tree* $T(E, I, \Delta, n)$ for a given a Mathias condition (E, I), Turing functional Δ, and $n \in \omega$ is defined as follows: $\lambda \in T(E, I, \Delta, n)$ and $\alpha \in T(E, I, \Delta, n)$ if α is a strictly increasing sequence of elements in I such that

$$(\forall F \subseteq \mathrm{ran}(\alpha^\#))(\forall w \geq n)(\Delta^{E\cup F}(w) \simeq 0).$$

Notice $T(E, I, \Delta, n)$ is clearly closed under prefixes and thus is a subtree of $I^{<\omega}$.

For our purposes, we only consider functionals Δ with range contained in $\{0,1\}$. The key properties of the extension tree are summarized in the following lemma (which appears as Lemma 3.2 of [7]).

Lemma 3.3.5. *The tree $T = T(E, I, \Delta, n)$ has the following properties*

1. *If T has an infinite path f, then $I' = \mathrm{ran}(f) \subseteq I$ satisfies*

$$(\forall F \subseteq I')(\forall w \geq n)(\Delta^{E \cup F}(w) \simeq 0).$$

2. *If $\alpha \in T$ is not terminal, then for all $x \in I$ such that $x > \mathrm{ran}(\alpha)$, $\alpha * x \in T$.*

3. *If $\alpha \in T$ is terminal, then there is a finite set $F \subseteq \alpha$ such that*

$$(\exists w \geq n)(\Delta^{E \cup F}(w) = 1).$$

4. *There is some $w \geq n$ with $\Delta^E(w) = 1$ if and only if $T(E, I, \delta, n) = \{\lambda\}$.*

Proof.

1. Suppose not. Then there is a set $F \subseteq I'$ and w witnessing $\Delta^{E \cup F}(w) \downarrow = 1$. Without loss of generality, assume F is finite. Let $\alpha \prec f$ be such that $F \subseteq \mathrm{ran}(\alpha^{\#})$. By construction, $\alpha \notin T$, a contradiction.

2. If $\alpha \in T$ is non-terminal, then there is some $\beta \in T$ such that $\beta^{\#} = \alpha$. So every $F \subseteq \mathrm{ran}(\alpha)$ is such that $\Delta^{E \cup F}(w) \simeq 0$ for all $w \geq n$. Thus for every $x \in I$ with $x > \mathrm{ran}(\alpha)$, $\alpha * x \in T$ because $\alpha * x$ is increasing and $(\alpha * x)^{\#} = \alpha$.

3. If $\alpha \in T$ is terminal, then for all $x \in I$ with $x > \mathrm{ran}(\alpha)$, $(\alpha * x) \notin T$. Thus, as $(\alpha * x)^{\#} = \alpha$, there are witnesses $F \subseteq \mathrm{ran}(\alpha)$ and $w \geq n$ with $\Delta^{F \cup N}(w) \downarrow = 1$.

4. For the if direction, note that if $T = \{\lambda\}$, then λ is terminal in T and the statement follows from item 3. For the only if direction, notice $\alpha \in T$ and $\alpha \neq \lambda$ implies there are no finite sets $F \subseteq \operatorname{ran}(\alpha^{\#})$ such that $\Delta^{E\cup F}(w) = 1$ for some $w \geq n$. But this occurs when $F = \emptyset$. Thus if $\alpha \neq \lambda$, then $\alpha \notin T$. Consequently, $T = \{\lambda\}$.

$\square$

If $T(E, I, \Delta, n)$ is well-founded then we may extend every $\alpha \in T$ to a terminal node β which has a witness $w \geq n$ to the statement

$$\exists F \subseteq \operatorname{ran}(\beta)(\Delta^{E\cup F}(w) \downarrow= 1).$$

We call the least such witness w the *label of* β, denoted $\operatorname{lb}(\beta)$, and use these to label every node in the tree.

Definition 3.3.6. Suppose an extension tree $T = T(E, I, \Delta, n)$ is well-founded. Beginning with the terminal nodes of T, we recursively define a function $\operatorname{lb} : T \to \omega \cup \{\infty\}$ assigning to each $\alpha \in T$ a *label* $\operatorname{lb}(\alpha)$. If $\alpha \in T$ is terminal, there is some $w \geq n$ and $F \subseteq \operatorname{ran}(\alpha)$ such that $\Delta^{E\cup F}(w) = 1$. Let $\operatorname{lb}(\alpha)$ be the least such witness w. If $\alpha \in T$ is not terminal, assume recursively that $\operatorname{lb}(\alpha * x)$ is defined for all $x \in I$ with $x > \operatorname{ran}(\alpha)$. If there is a number w such that $\operatorname{lb}(\alpha * x) = w$ for infinitely many x, let $\operatorname{lb}(\alpha)$ be the least such w. Otherwise, let $\operatorname{lb}(\alpha) = \infty$. We call lb a *labeling* of T and say $\operatorname{lb}(\alpha)$ is *finite* if $\operatorname{lb}(\alpha) \in \omega$.

Note the label of $\operatorname{lb}(\alpha)$ is finite if and only if infinitely many immediate successors of α share this same label. If $\operatorname{lb}(\alpha) = \infty$ then either infinitely many immediate successors of α share this label or for any successor $\alpha * x \in T$, the set $\{\alpha * y \in T :$

$\mathrm{lb}(\alpha * y) = \mathrm{lb}(\alpha * x)\}$ is finite. We prune T to retain only this information: the *labeled subtree* of T is defined so that the immediate successors of any node will either all share the same label, or each have a distinct finite label.

Definition 3.3.7. If $T = T(E, I, \Delta, n)$ is a well-founded extension tree with labeling lb, we define the *labeled subtree* T^L of T recursively as follows. Place $\lambda \in T^L$. Now, assume $\alpha \in T^L$. If $\mathrm{lb}(\alpha)$ is finite, place $\alpha * x \in T^L$ for all x such that $\mathrm{lb}(\alpha * x) = \mathrm{lb}(\alpha)$. If $\mathrm{lb}(\alpha) = \infty$ and infinitely many immediate successors of α have label ∞, place each such successor into T^L. Otherwise, $\mathrm{lb}(\alpha) = \infty$ and there are infinitely many finite labels w such that $\mathrm{lb}(\alpha * x) = w$ for some $\alpha * x \in T$. In this case, for each such label w place $\alpha * x$ into T^L if x is least such that $\mathrm{lb}(\alpha * x) = w$.

Note that if lb is a labeling of an extension tree T, then its restriction to the labeled subtree T^L is a labeling of T^L. When the domain of lb is clear from context, we will use lb to refer to either the labeling of T or T^L.

The utility of this framework reveals itself when Δ is treated as a backward functional in a potential (strong) computable reduction. If $\mathrm{lb}(\lambda) = w$, the idea is to continue our construction in such a way that w will prevent Δ^X from being a solution to the original instance. Here X is any infinite set compatible with the Mathias condition (E, I). For example, if Δ^E is constructing a chain in our stable poset, we may make w isolated to prevent Δ^X from being a solution. To force $w \in \Delta^X$ we search for a terminal string $\alpha \in T^L$. Since it must also have label w, there will be a set $F \subseteq \mathrm{ran}(\alpha)$ such that the Mathias condition $(E \cup F, \{x \in I : x > F\})$ will force $w \in \Delta^X$ for any compatible X. We then will have diagonalized Δ in the construction. The challenge will be finding $\alpha \in T^L$ such that $E \cup \mathrm{ran}(\alpha)$ has the desired structure in the original instance (e.g., $E \cup \mathrm{ran}(\alpha)$ is a chain of small elements). If our instance involves a stable limit behavior (e.g., being small or isolated), then we will be able to find some one-point extension of a given $\alpha \in T^L$ as the row below α is infinite. At some point, the relationship of the elements in $\operatorname{ran}(\alpha)$ will have stabilized with respect to sufficiently large elements in the row below α. So long as we find a non-terminal $\alpha \in T^L$ with a finite label, this approach will succeed. If instead, $\operatorname{lb}(\lambda) = \infty$ and our search through T^L takes us to a pre-leaf α with $\operatorname{lb}(\alpha) = \infty$, then the labels of the successors $\alpha * x$ will all be distinct and finite. In this case, we may not be able to find a suitable x and label w to simultaneously extend our condition via $\operatorname{ran}(\alpha * x)$ while diagonalizing Δ with w. This will only occur if the limit behavior of w directly determines the limit behavior of x, and in this way, we will have gained some control over the global structure of the computed instance in the reduction. Of course, at any point in this search, we may simply find an infinite set $R \subseteq I$ which will be suitable to add to $\mathcal{D}$.

We are now ready to prove that there is a stable poset G witnessing that $\mathsf{SCAC} \not\leq_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{ord}}$. To summarize our approach, for every triple of functionals (Φ, Ψ_0, Ψ_1) we seek to build an infinite chain C_Φ and infinite chain A_Φ in Φ^G such that either $\Psi_0^{C_\Phi}$ is not a solution of G or $\Psi_1^{A_\Phi}$ is not a solution of G. We will first attempt to make progress on the chain C_Φ, and then attempt to make progress on the antichain A_Φ. If both attempts fail, we will encounter a suitable infinite set R to add to a collection $\mathcal{D}$. On other stages of the construction, we will ensure G is sufficiently generic over the elements of $\mathcal{D}$ so that G has no $(G \oplus R)$-computable solutions for these $R \in \mathcal{D}$. We will conclude by showing that if progress cannot be made on the chain or antichain, and no such R can be found, then Φ^G is either not stable or not ω-ordered. The latter will be done by exemplifying a pair $a <_\omega b$ with $b <_{\Phi^G} a$.

Theorem 3.3.8. *There is a stable poset G and a collection of infinite sets $\mathcal{D}$ such that for any $R \in \mathcal{D}$, no $(G \oplus R)$-computable infinite set is a chain or antichain of G. Moreover, any stable ω-ordered poset $\widehat{G}$ computable from G has either: an infinite chain or infinite antichain computable in $(G \oplus R)$ for some $R \in \mathcal{D}$, or an infinite chain or infinite antichain that computes no infinite chain or infinite antichain in G.*

Proof. To construct the required objects we build the following:

- a sequence $p_0 \leq_{\mathbb{P}} p_1 \leq_{\mathbb{P}} \cdots$ of $\mathbb{P}$-conditions;
- two sequences of finite sets $C_{\Phi,0} \subseteq C_{\Phi,1} \subseteq \cdots$, and $A_{\Phi,0} \subseteq A_{\Phi,1} \subseteq \cdots$ for each functional Φ;
- a decreasing sequence of infinite sets $R_0 \supseteq R_1 \supseteq \cdots$ such that $C_{\Phi,s} < R_s$ and $A_{\Phi,s} < R_s$ for all s and Φ; and
- an increasing sequence of finite families $D_0 \subseteq D_1 \subseteq \cdots$ of infinite subsets of ω.

In the end $G = \bigcup_{n\in\omega} \pi^{p_n}$, $C_\Phi = \bigcup_s C_{\Phi,s}$, and $A_\Phi = \bigcup_s A_{\Phi,s}$, for each functional Φ, and $\mathcal{D} = \bigcup_s D_s$. We need to make G sufficiently generic over $\mathcal{D}$, and if possible, to ensure C_Φ and A_Φ are respectively an infinite chain and infinite antichain in Φ^G, one of which computes no infinite chain or infinite antichain in G. Toward these ends, we satisfy the following requirements for each $s \in \omega$ and all Turing functionals Φ, Ψ_0 and Ψ_1.

$\mathscr{G}_s$: G is sufficiently R-generic for every $R \in D_s$.

$\mathscr{D}_{\Phi,\Psi_0,\Psi_1}$: If Φ^G is an stable ω-ordered poset, then either Φ^G has an infinite chain or infinite antichain computable in $(G \oplus R)$ for some $R \in \mathcal{D}$, or there is an infinite chain C_Φ and infinite antichain A_Φ in Φ^G such that one of $\Psi_0^{C_\Phi}$ or $\Psi_1^{A_\Phi}$ is not an infinite chain or infinite antichain in G.

Distribute the stages of the construction in such a way so that each $\mathscr{G}$ requirement is addressed infinitely often and each $\mathscr{D}$ requirement is addressed once. By Lemma 3.3.1, the $\mathscr{G}$ requirements will ensure that G has no infinite $(G \oplus R)$-computable chain or antichain for each $R \in \mathcal{D}$. By Lachlan's Disjunction (Proposition 3.3.2), the $\mathscr{D}$ requirements will ensure that for each Φ, one of C_Φ or A_Φ diagonalizes Φ^G with respect to any backward functional Ψ.

Construction. To begin, let $p_0 \in \mathbb{P}$ be any condition such that $a^p : |\pi^p| \to \{S, I\} \times (\omega \upharpoonright |\pi^p| + 1)$. So G will be a stable poset of the small type. Let $R_0 = D_0 = \emptyset$ and let $C_{\Phi,0} = A_{\Phi,0} = \emptyset$ for all functionals Φ. Suppose we are at stage s and given $p_s, C_{\Phi,s}, A_{\Phi,s}$ for all Φ, and D_s. Assume inductively that if $C_{\Phi,s}$ or $A_{\Phi,s}$ is nonempty for some Φ, then p_s forces Φ^G is an stable ω-ordered poset and that $C_{\Phi,s}$ and $A_{\Phi,s}$ are respectively a chain of small elements and an antichain of isolated elements in Φ^G. At the conclusion of the stage, if any of p_{s+1}, $C_{\Phi,s+1}$, $A_{\Phi,s+1}$, R_{s+1}, and D_{s+1} have not been explicitly defined, let them equal p_s, $C_{\Phi,s}$, $A_{\Phi,s}$, R_s, and D_s respectively.

$\mathscr{G}$ requirements. These are satisfied in a systematic and straightforward way. Let n be sufficiently large to satisfy Lemma 3.3.1. Suppose $s > t$ is the $\langle \ell, m\rangle$th stage dedicated to $\mathscr{G}_t$. If $\ell > |D_s|$, do nothing. Else, let R be the ℓth set in D_s and W be the mth $\Sigma^0_n(R)$ set. If G has an extension $q \in W$, set $p_{s+1} = q$ and proceed to the next stage. Otherwise, do nothing.

$\mathscr{D}$ requirements. Suppose stage s is dedicated to requirement $\mathscr{D}_{\Phi,\Psi_0,\Psi_1}$. We begin with a few initial extensions of p_s, $C_{\Phi,s}$, $A_{\Phi,s}$ and R_s before attempting to make

progress on the chain $C_{\Phi,s}$ with an extension tree. If this fails, we repeat this search with a second extension tree to make progress on $A_{\Phi,s}$. Throughout this process, if at any point we encounter an infinite set R which will contain only finitely many small or finitely many isolated elements of Φ^G, we add R to D_s and proceed to the next stage. At the end, if each of these approaches fails, it will contradict that Φ^G is ω-ordered.

Initialization.

Begin by extending p_s if necessary to a condition forcing that Φ^G is a stable ω-ordered poset. If no such extension exists, $\mathscr{D}_{\Phi,\Psi_0,\Psi_1}$ is vacuously satisfied and we proceed to the next stage. Next, search for an extension q forcing either of the following statements:

- The number of small elements in $\Phi^G \cap R_s$ is finite.
- The number of isolated elements in $\Phi^G \cap R_s$ is finite.

If either holds, then there is set D cofinite in R_s such that D will contain only small or only isolated elements of Φ^G. Thus D can be computably thinned to an infinite chain or infinite antichain in Φ^G. Hence D is an $(R_s \oplus P)$-computable set which computes an infinite chain or antichain of Φ^G. So we set $D_{s+1} = D_s \cup \{R_s\}$ and $p_{s+1} = q$ and proceed to the next stage, having satisfied $\mathscr{D}_{\Phi,\Psi_0,\Psi_1}$.

If there is no such condition, we conclude that R_s will contain infinitely many small elements and infinitely many isolated elements of Φ^G. Thus move to an extension $q \leq_{\mathbb{P}} p_s$ and a triple of sets (C, A, J) such that

- $C_{\Phi,s} \subseteq C \subseteq C_{\Phi,s} \cup R_s$ and C is a finite chain in Φ^{π^q} with $|C_{\Phi,s}| < |C|$;
- $A_{\Phi,s} \subseteq A \subseteq A_{\Phi,s} \cup R_s$ and A is a finite antichain in Φ^{π^q} with $|A_{\Phi,s}| < |A|$;

- J is the infinite set $\{x \in R_s : x > A \wedge x > C\}$; and
- q forces that each $x \in C$ is small in Φ^G, each $x \in A$ is isolated in Φ^G, and for all $x > |\pi^q|$, $C \cup \{x\}$ and $A \cup \{x\}$ are respectively a chain and antichain in Φ^G.

Note we have extended $C_{\Phi,s}$ and $A_{\Phi,s}$ by a finite amount, ensuring that C_Φ and A_Φ will be infinite. Additionally, choose C and A large enough that Ψ_0^C and Ψ_1^A have size at least two. If this is not possible, then one of $\Psi_0^{C_\Phi}$ or $\Psi_1^{A_\Phi}$ must be finite. In this case, we set $C_{\Phi,s+1} = C$, $A_{\Phi,s+1} = A$ and $R_{s+1} = J$ and proceed to the next stage having made progress on at least one of the chain or the antichain and thereby satisfied $\mathscr{D}_{\Phi,\Psi_0,\Psi_1}$.

Otherwise, we seek to extend p_s, C, A and J to p_{s+1}, $C_{\Phi,s+1}$, $A_{\Phi,s+1}$ and R_{s+1} via tree labeling. We first attempt to make progress on C with an extension tree for Ψ_0. We then attempt to make progress on A with an extension tree for Ψ_1. Throughout both attempts, if at any point an infinite set R is found which will only contain finitely many small or finitely many isolated elements of Φ^G, then we will proceed to the next stage after setting $D_{s+1} = D_s \cup \{R\}$. Unless explicitly defined otherwise, we let $p_{s+1} = p_s$, $C_{\Phi,s+1} = C$, $A_{\Phi,s+1} = A$, and $R_{s+1} = J$.

Making progress on the chain.

Construct the extension tree $T(C, J, \Psi_0, |\pi^q|)$. If this tree has a path f, let $R_{s+1} = \operatorname{ran}(f)$ and proceed to the next stage. Note this satisfies $\mathscr{D}_{\Phi,\Psi_0,\Psi_1}$ as the set defined by $\Psi_0^{C_\Phi}$ will be finite. Indeed, for sufficiently large n the oracle use of $\Psi_0^{C_\Phi}(n)$ is contained in $C \cup F$ for some finite $F \subseteq \operatorname{ran}(f)$ and the definition of the extension tree ensures $\Psi_0^{C\cup F}(n) \simeq 0$.

If $T(C, J, \Psi_0, |\pi^q|)$ is well-founded, construct a labeling $\mathrm{lb} : T \to \omega \cup \{\infty\}$ and the corresponding labeled subtree T_C^L. We aim to determine a finite sequence of $\mathbb{P}$

conditions $q \geq_{\mathbb{P}} q_0 \geq_{\mathbb{P}} q_1 \geq_{\mathbb{P}} \cdots \leq_{\mathbb{P}} p_{s+1}$ alongside a sequence of strings $\alpha_0 \preceq \alpha_1 \preceq \cdots \preceq \alpha_n \preceq \alpha$ in T_C^L in which α is terminal and has an appropriate $F \subseteq \operatorname{ran}(\alpha_n)$ to extend C to $C_{\Phi,s+1}$.

Let $\alpha_0 = \lambda$. If $\operatorname{lb}(\alpha_0) = w$, find an extension $q_0 \leq_{\mathbb{P}} q$ such that $a^{q_0}(w) = (I, |\pi^{q_0}|)$ if Ψ_0^C contains two comparable elements in π^q or $a^{q_0}(w) = (S, |\pi^{q_0}|)$ otherwise. If $\operatorname{lb}(\alpha_0) = \infty$, set $q_0 = q$. Assume we have obtained a condition q_n and nonterminal string $\alpha_n \in T_C^L$ such that $\operatorname{ran}(\alpha_n)$ is a chain in $\Phi^{\pi^{q_n}}$ and q_n forces each $x \in \operatorname{ran}(\alpha_n)$ small in Φ^G. To extend α_n, we consider two cases depending on whether or not the label of α_n agrees with the label of each of its successors.

Case 1: $\operatorname{lb}(\alpha_n) = \operatorname{lb}(\alpha_n * x)$ for each x with $\alpha * x \in T_C^L$. As each element of $\operatorname{ran}(\alpha_n)$ is forced small, there will be some t such that for all $x > t$ with $\alpha_n * x \in T_C^L$, the set $\operatorname{ran}(\alpha_n) \cup \{x\}$ will be a chain in Φ^G. Search for a condition $q_{n+1} \leq_{\mathbb{P}} q_n$ and x such that $\alpha_n * x \in T_C^L$, $\operatorname{ran}(\alpha_n * x)$ is a chain in $\Phi^{\pi^{q_{n+1}}}$, and q_{n+1} forces x small in Φ^G. Let $\alpha_{m+1} = \alpha_n * x$ and proceed. If there is no such condition q_{n+1}, then q_n forces that the row below α_n, $R = \{x : \alpha_n * x \in T_C^L\}$, will contain only finitely many small elements of Φ^G. So add R to D_s, set $p_{s+1} = q_n$, and proceed to the next stage having satisfied the requirement.

Case 2: $\operatorname{lb}(\alpha_n) \neq \operatorname{lb}(\alpha_n * x)$ for some $\alpha_n * x \in T_C^L$. Notice this case can only occur if $\operatorname{lb}(\alpha_n) = \infty$ and each of its immediate successors $\alpha_n * x$ have distinct finite labels. Note again that any condition $q_{n+1} \leq_{\mathbb{P}} q_n$ will guarantee $\operatorname{ran}(\alpha_n * x)$ is a chain in Φ^G for sufficiently large x. Search for a condition $q_{n+1} \leq_{\mathbb{P}} q_n$, and an x and w such that $\alpha_n * x \in T_C^L$, $\operatorname{lb}(\alpha_n * x) = w$, $\operatorname{ran}(\alpha_n * x)$ is a chain in $\Phi^{\pi^{q_{n+1}}}$, and that either

- q_{n+1} forces x small in Φ^G with $a^{q_{n+1}}(w) = (I, |\pi^{q_{n+1}}|)$ if Ψ_0^C has two comparable elements in π^q; or

- q_{n+1} forces x small in Φ^G with $a^{q_{n+1}}(w) = (S, |\pi^{q_{n+1}}|)$ if Ψ_0^C lacks two comparable elements in π^q.

Let $\alpha_{n+1} = \alpha_n * x$ and proceed. If no such condition exists, let

$$R_C = \{x : \alpha_n * x \in T_C^L \wedge \operatorname{ran}(\alpha_n * x) \text{ will be a chain in } \Phi^G \wedge \operatorname{lb}(\alpha * x) > |\pi^{q_n}|\}$$

and set $R_C^L = \{\langle x, w\rangle : x \in R_C \wedge \operatorname{lb}(\alpha_n * x) = w\}$. Note R_C is an infinite subset of the row below α_n. For every $\langle x, w\rangle \in R_C^L$, if $q' \leq_{\mathbb{P}} q_n$ and $a^{q'}$ assigns the required limit behavior to w (depending on whether or not Ψ_0^C has two comparable elements in π^q) then q' cannot force x small in Φ^G. Hence, q' must force x isolated in Φ^G. In this case, we have failed to extend α_n but have gained some control on the limit behavior of $x \in R_C$. So we must attempt to make progress on the antichain.

If instead we successfully extended $\alpha_0 \preceq \cdots \preceq \alpha_n$ to a terminal string $\alpha \in T_A^L$ at condition q_n, then $\operatorname{lb}(\alpha)$ is finite, say with value w. In this case there is some $F \subseteq \operatorname{ran}(\alpha)$ such that $\Psi_0^{C\cup F}(w) \downarrow = 1$, and q_n forces that every element in the chain $C \cup F$ is small in Φ^G. Moreover, q_n will make w isolated in G if $\Psi_0^{C\cup F}$ contains two comparable elements of π^q, and otherwise make w small. Either way, any infinite chain X extending $C \cup F$ in Φ^G will have that Ψ_0^X is not an infinite chain or antichain of G. We set $p_{s+1} = q_n$, $C_{\Phi,s+1} = C \cup F$ and $R_{s+1} = \{x \in J : x > C \cup F \cup A\}$ and proceed to the next stage, having made progress on the chain.

If at any point, we fail to extend some α_n then we either found a set R to add to D_s or we are at a condition q_n with the sets R_C and R_C^L. With newfound leverage, we repeat the search in an extension tree for the antichain A.

Making progress on the antichain.

Construct the extension tree $T(A, R_C, \Psi_1, |\pi^{q_n}|)$. As with the previous extension tree, if $T(A, R_C, \Psi_1, |\pi^{q_n}|)$ has an infinite path f, set $p_{s+1} = q_n$, and $R_{s+1} = \text{ran}(f)$ and proceed to the next stage. Otherwise, $T(A, R_C, \Psi_1, |\pi^{q_n}|)$ is well-founded. Form a labeling lb of $T(A, R_C, \Psi_1, |\pi^{q_n}|)$ and the corresponding labeled subtree T^L_A. We again aim to construct a finite sequence of conditions $q_n \geq_{\mathbb{P}} q_{n+1} \geq_{\mathbb{P}} \geq_{\mathbb{P}} p_{s+1}$ and a terminal string $\alpha \in T^L_A$ containing a suitable $F \subseteq \text{ran}(\alpha)$ with which to extend A to $A_{\Phi,s+1}$.

We build a sequence $\alpha_1 \preceq \alpha_2 \preceq \cdots \preceq \alpha_m \preceq \alpha \in T^L_A$ as above while highlighting the differences. Let $\alpha_1 = \lambda$. If $\text{lb}(\alpha_1) = w$, move to an extension $q_{n+1} \leq_{\mathbb{P}} q_n$ with $a^{q_{n+1}}(w) = (I, |\pi^{q_{n+1}}|)$ if Ψ^A_1 contains two comparable elements in π^q, or $a^{q_{n+1}}(w) = (S, |\pi^{q_{n+1}}|)$ otherwise. If $\text{lb}(\alpha_0) = \infty$, set $q_{n+1} = q_n$. Assume we have constructed α_m and found condition q_{n+m} such that $\text{ran}(\alpha_m)$ is an antichain in $\Phi^{\pi^{q_{n+m}}}$ and q_{n+m} forces each element in $\text{ran}(\alpha_m)$ isolated in Φ^G. We again consider two cases to extend α_m by one element.

Case 1: $\text{lb}(\alpha_m) = \text{lb}(\alpha_m * x)$ for all x with $\alpha_m * x \in T^L_A$. Since the elements of $\text{ran}(\alpha_m)$ are forced isolated in Φ^G, we conclude $\text{ran}(\alpha * x)$ will be an antichain in Φ^G for sufficiently large x with $\alpha * x \in T^L_A$. Thus, if there is an x and condition q_{n+m+1} with $\alpha * x \in T^L_A$, $\text{ran}(\alpha * x)$ is an antichain in $\Phi^{\pi^{q_{n+m+1}}}$, and such that q_{n+m+1} forces x isolated in Φ^G, let $\alpha_{m+1} = \alpha_m * x$ and proceed. Otherwise, q_{n+m} forces the row R below α_m in T^L_A to contain only finitely many isolated elements of Φ^G. So we set $p_s = q_{n+m}$, $R_{s+1} = R_C$, and $D_{s+1} = D_s \cup \{R\}$ and conclude this stage of the construction.

Case 2: $\text{lb}(\alpha_m) \neq \text{lb}(\alpha_m * x)$ for some $\alpha_m * x \in T^L_C$. Then $\text{lb}(\alpha_m) = \infty$ and for sufficiently large x, q_{n+m} forces $\text{ran}(\alpha_m * x)$ to be an antichain in Φ^G. Search for a condition $q_{n+m+1} \leq_{\mathbb{P}} q_{n+m}$, and an x and w such that $\alpha_m * x \in T^L_A$, $\text{lb}(\alpha_m * x) = w$, $\text{ran}(\alpha_m * x)$ is an antichain in $\Phi^{\pi^{q_{n+m+1}}}$, and that either

- q_{n+m+1} forces x isolated in Φ^G with $a^{q_{n+m+1}}(w) = (I, |\pi^{q_{n+m+1}}|)$ if Ψ_1^A has two comparable elements in π^q; or

- q_{n+m+1} forces x isolated in Φ^G with $a^{q_{n+m+1}}(w) = (S, |\pi^{q_{n+m+1}}|)$ if Ψ_1^A lacks two comparable elements in π^q.

Let $\alpha_{m+1} = \alpha_m * x$ and proceed. If no such condition exists, let

$$R_A = \{x : \alpha_m * x \in T_C^L \wedge \operatorname{ran}(\alpha_m * x) \text{ will be an antichain in } \Phi^G \wedge \operatorname{lb}(\alpha * x) > |\pi^{q_{n+m}}|\}.$$

and set $R_A^L = \{\langle x, w\rangle : x \in R_A \wedge \operatorname{lb}(\alpha * x) = w\}$. As before R_A is an infinite subset of the row below α_m. And similar to R_C^L, we have that for any $\langle x, w\rangle \in R_A^L$, if $q' \leq_{\mathbb{P}} q_{n+m}$ and assigns the correct limit behavior to w, then q' forces x small in Φ^G. In this situation, we will make progress on the antichain by finding a finite set $F \subseteq R_A$ to extend A.

If we successfully extended $\alpha_1 \preceq \cdots \preceq \alpha_m$ to a terminal string $\alpha \in T_A^L$ at condition q_{n+m}, then $\operatorname{lb}(\alpha) = w$ for some w. There is a set $F \subseteq \operatorname{ran}(\alpha)$ such that $\Psi_1^{A\cup F}(w) \downarrow= 1$. Furthermore, q_{n+m} forces $A \cup F$ to be an antichain of isolated elements in Φ^G and assigns w the correct limit behavior to ensure any infinite antichain X extending $A \cup F$ does not define via Ψ_1 an infinite chain or antichain of G. Thus we have made progress on the antichain and set $p_{s+1} = q_{n+m}$, $A_{\Phi,s+1} = A \cup F$, and $R_{s+1} = \{x \in R_C : x > A \cup F \cup C\}$ before proceeding to the next stage.

If we failed to extend some α_m, then we either found a set R to add to D_s or we are at condition q_{n+m} with the sets R_A and R_A^L as well as the sets R_C and R_C^L. Search for a finite set $F \subseteq R_A$ such that $\Psi_1^{A\cup F}(a) \downarrow= \Psi_1^{A\cup F}(b) \downarrow= 1$ for two fresh elements $a, b > |\pi^{q_{n+m}}|$. If we find such an F, a, and b, we claim there is an extension $q' \leq_{\mathbb{P}} q_{n+m}$ which forces $A \cup F$ to be an antichain of isolated elements in Φ^G and guarantees that $\Psi_1^{A \cup F}$ contains two incomparable but small elements or two comparable but isolated elements of G. Thus we set $p_{s+1} = q'$, $A_{\Phi,s+1} = A \cup F$, and $R_{s+1} = \{x \in R_A : x > A \cup F \cup C\}$, and proceed to the next stage having made progress on the antichain. If there is no such set $F \subseteq R_A$, then set $R_{s+1} = R_A$ and $p_{s+1} = q_{n+m}$. This guarantees that any infinite antichain X extending A will have Ψ_1^X finite so we make progress on the antichain in this case as well. This concludes the construction.

It remains to prove the claim and verify the construction. We first prove the claim. The idea is to show that if such a condition q' does not exist, then Φ^G is not ω-ordered.

Claim. *There is a condition $q' \leq_{\mathbb{P}} q_{n+m}$ which forces that $A \cup F$ is an antichain of isolated elements in Φ^G, and there are two elements $a, b \in \Psi_1^{A \cup F}$ such that either*

1. *$\{a, b\}$ is an antichain in $\pi^{q'}$, and $a^{q'}$ sets both a and b to be small, or*
2. *$\{a, b\}$ is a chain in $\pi^{q'}$, and $a^{q'}$ sets both a and b to be isolated.*

Proof of the claim. Recall $q \leq_{\mathbb{P}} q_{n+m}$ forces that A contains only isolated elements of Φ^G and that for any $x > |\pi^q|$, $A \cup \{x\}$ is an antichain in Φ^G. So it suffices to show the existence of a condition $q' \leq_{\mathbb{P}} q_{n+m}$ which makes F an antichain of isolated elements in Φ^G and for which one of statements 1 or 2 hold.

To begin, note that $R_A \subseteq R_C$ by construction and moreover, $R_A^L \subseteq R_C^L$. If not, then there is an $x \in R_A$ such that $(x, w) \in R_A^L$ and $(x, w') \in R_C^L$. Extending q_{n+m} to a condition in which w and w' have the correct limit behaviors forces x to be isolated by its membership in R_C and small by its membership in R_A, a contradiction. Thus we have complete control over the limit behavior of the elements of F as $F \subseteq R_A \subseteq R_C$.

To elaborate, for each $x \in F$ there is a w such that $\langle x, w \rangle \in R_A^L$. Any condition $r \leq_{\mathbb{P}} q_{n+m}$ determines whether x will be small or isolated in Φ^G by the limit behavior it assigns w (depending on the structure of Ψ_0^C and Ψ_1^A). So we may ensure the elements of F are isolated via the labels of its elements. Unless otherwise mentioned, we now only consider labels w corresponding to $x \in F$.

The key fact to notice is that for any condition r, we may freely adjust the limit behaviors as needed of a, b and the labels w by using a parallel condition r'. Since the limit behaviors a^r assigns to a, b and each w have no effect on the local structure of π^r, we may take a parallel condition r' assigning limit behavior via $a^{r'}$ in the way we need. This will not affect the local structure of F as $\pi^r = \pi^{r'}$ so $\Phi^{\pi^r} = \Phi^{\pi^{r'}}$. Thus, if no q' exists, we can conclude that this failure is not witnessed by limit behavior but instead locally. For example, if statement 1 fails this would mean that if $\{a, b\}$ is an antichain in G, then F cannot itself be an antichain in Φ^G.

We will now show by contradiction that either statement 1 or 2 must hold. Since the way each label w affects the corresponding element in F depends on the structure of Ψ_0^C and Ψ_1^A, we have four cases to consider. The concern will be when a and b are themselves labels of F.

Case 1: Ψ_0^C and Ψ_1^A both have two comparable elements in π^q. Here if $\langle x, w \rangle \in R_A^L$, then any $r \leq_{\mathbb{P}} q_{n+m}$ with w isolated forces x both small and isolated in Φ^G. So case 1 cannot obtain.

Case 2: Ψ_0^C and Ψ_1^A both lack two comparable elements in π^q. Here if $\langle x, w \rangle \in R_A^L$, then any $r \leq_{\mathbb{P}} q_{n+m}$ with w small forces x both small and isolated in Φ^G. So case 2 cannot obtain.

Case 3: Ψ_0^C lacks two comparable elements in π^q but Ψ_1^A does not. Here if $\langle x, w\rangle \in R_A^L$, then any $r \leq_{\mathbb{P}} q_{n+m}$ which has w isolated forces x small in Φ^G because $x \in R_A$ and any $r \leq_{\mathbb{P}} q_{n+m}$ with w small forces x isolated in Φ^G because $x \in R_C$.

In this case, we show statement 1 must hold. Suppose not: that is, that there is no condition q' which forces F to be an antichain of isolated elements in Φ^G while having $\{a, b\}$ a chain of isolated elements in $\pi^{q'}$. As mentioned above, this cannot occur due to limit behavior, but instead happens due to local behavior. Specifically, any condition $r \leq_{\mathbb{P}} q_{n+m}$ which has $\{a, b\}$ an antichain in π^r forces $x <_{\Phi^G} y$ for some $x, y \in F$. Choose a condition r which makes the label of x small, forcing x isolated, and makes the label of y isolated, forcing y small. Note that as $a \mid_{\pi^r} b$, there are no concerns over whether or not a and b label x and y. As r forces $x <_{\Phi^G} y$ with y small, x must be small in Φ^G. Then x will be both small and isolated, a contradiction

Case 4: Ψ_0^C has two comparable elements in π^q but Ψ_1^A does not. Here if $\langle x, w\rangle \in R_A^L$, then any $r \leq_{\mathbb{P}} q_{n+m}$ which has w isolated forces x isolated in Φ^G because $x \in R_C$ and any $r \leq_{\mathbb{P}} q_{n+m}$ with w small forces x small in Φ^G because $x \in R_A$.

Here we show statement 2 must hold. Suppose not: then any condition $r \leq_{\mathbb{P}} q_m$ setting $a <_G b$ forces $x <_{\Phi^G} y$ for some $x, y \in F$. From this, we conclude $\langle x, a\rangle, \langle y, b\rangle \in R_A^L$. That is, a and b are the labels corresponding to x and y. To see this, note first that if $\langle x, b\rangle, \langle y, a\rangle \in R_A^L$, then any condition setting b isolated and a small, would in turn force x isolated and y small. But then the fact that $x <_{\Phi^G} y$ would imply x is also small in

Φ^G, a contradiction. Next suppose b is not the label for y. We can then force x isolated and y small via their labels, without affecting the limit behavior of b, and obtain a similar contradiction. *Mutatis mutandis*, we have that a is the label for x.

We also note that any condition setting $b <_G a$ forces $y <_{\Phi^G} x$. This follows from the structure of T_A^L since every element in R_A corresponds to a distinct label. Since $b <_G a$ forces some $w <_{\Phi^G} z$ with $w, z \in F$, we have that b is the label for w and a the label for z. So $w = y$ and $z = x$.

Summarizing, setting $a <_G b$ or $b <_G a$ forces $x <_{\Phi^G} y$ or $y <_{\Phi^G} x$ respectively. But Φ^G is ω-ordered, so only one of $x <_{\Phi^G} y$ or $y <_{\Phi^G} x$ is valid. Assume it is the former, and move to a condition $r \leq_{\mathbb{P}} q_{n+m}$ such that $b <_G a$. Then r forces that $y <_{\Phi^G} x$ with $x <_\omega y$, contradicting that Φ^G is ω-ordered.

In every case, we show there must be a condition q' for which statement 1 or 2 holds. This completes the proof of the lemma.

Verification. Let G be the generic poset and $\mathcal{D}$ the collection of infinite sets resulting from the construction. If Φ is a functional such that Φ^G is an stable ω-ordered poset, then either it has a $(G \oplus R)$-computable infinite chain or infinite antichain X for some $R \in \mathcal{D}$, or two infinite sets C_Φ and A_Φ were constructed. In the first case, as each $\mathscr{G}$-requirement was satisfied, we have by Lemma 3.3.1 that Ψ^X is not an infinite chain or infinite antichain of G for any functional Ψ. In the second case, C_Φ and A_Φ are respectively a chain and antichain in Φ^G. Furthermore, for every pair of functionals (Ψ_0, Ψ_1) one of $\Psi_0^{C_\Phi}$ or $\Psi_1^{A_\Phi}$ is guaranteed not to be an infinite chain or infinite antichain of G by the $\mathscr{D}$-requirements. Applying Lachlan's Disjunction

(Proposition 3.3.2) yields that for some $X \in \{C_\Phi, A_\Phi\}$, Ψ^X is not an infinite chain or infinite antichain of G for any functional Ψ. Thus G is the desired stable poset and $\mathcal{D}$ is the desired collection of infinite sets. □

Corollary 3.3.9. $\mathsf{SCAC} \not\leq_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{ord}}$

Proof. This is witnessed by any stable poset satisfying Theorem 3.3.8. □

Note the stable poset G built in the proof of Theorem 3.3.8 is of the small type, so we obtain $\mathsf{SCAC}^{\mathsf{small}} \not\leq_{\mathsf{sc}} \mathsf{SCAC}^{\mathsf{ord}}$. More generally, we obtain the following in view of Corollary 3.1.14.

Corollary 3.3.10. *The principle* $\mathsf{SCAC}^{\mathsf{ord}}$ *is not strongly computably equivalent to any of the following principles:* $\mathsf{SCAC}^{\mathsf{small}}$, $\mathsf{SCAC}^{\mathsf{large}}$, *and* $\mathsf{SCAC}^{\mathsf{type}}$.

Chapter 4

Determining unique colorability in graphs and hypergraphs

In this chapter, we extend results of Davis, Hirst, Pardo, and Ransom [3] on proper k-colorings of hypergraphs. In particular, we initiate the study of unique colorability in hypergraphs and obtain a sorting principle equivalent to ATR_0. We then give preliminary results from joint work with Jeffry L. Hirst on the unique colorability of graphs and hypergraphs.

4.1 Proper colorability in hypergraphs

A *hypergraph* is a pair of sets $H = (V, E)$ with $E \subseteq \mathcal{P}(V)$. Intuitively, V is a set of vertices and E is a collection of edges each of which contain any number, finite or infinite, of the vertices. A *graph* is a hypergraph in which every edge has cardinality 2. We call two vertices u and v *adjacent* if $u, v \in e$ for some $e \in E$. For our purposes, we assume $V \subseteq \mathbb{N}$ and $E \subseteq \mathbb{N}\times\mathbb{N}$ encodes the countable collection $\{e_0, e_1, \dots\}$ of edges by

$e_m = \{n : \langle n, m\rangle \in \mathbb{N}\}$. We also confuse each integer k with the set $\{0, 1, \ldots, k-1\}$.

A *k-coloring* of a graph $G = (V, E)$ is a function $c : V \to k$ such that for all $u, v \in V$, if there is an edge $e \in E$ with $e = \{u, v\}$, then $c(u) \neq c(v)$. That is, if u and v are adjacent in G, they have different colors. Now in a hypergraph, there may be edges of size greater than k, so we allow for a weaker notion of a vertex coloring. Specifically, we say a function c is a *proper* k-coloring of a hypergraph $H = (V, E)$ if $c : V \to k$ and c is non-constant on every edge $e \in E$ with cardinality greater than 1. Directly generalizing k-colorings on a graph leads to a *strong* k-coloring of a hypergraph, i.e., a coloring c which is injective on every edge $e \in E$.

We say a graph G is *k-colorable* if there exists a k-coloring of its vertices. We say a hypergraph H is *properly k-colorable* if there exists a proper k-coloring of its vertices.

Davis, Hirst, Pardo, and Ransom [3] initiated the study of k-colorability of hypergraphs in reverse mathematics. In particular, they proved in RCA_0 that for a given k, determining which hypergraphs in an infinite sequence admit a proper k-coloring is equivalent to Π^1_1-CA_0 (see Theorem 6 of [3]). We have shown that in fact, their result can be modified to obtain a principle equivalent to ATR_0 over RCA_0. To do this, we restrict the question of determining proper k-colorability to determining *unique* proper k-colorability. Now, as any proper k-coloring defines other distinct proper k-colorings by permuting the colors, we address only 2-colorings and define the following.

Definition 4.1.1. We say $H = \langle V, E\rangle$ has a *unique proper 2-coloring* $f : V \to 2$ if and only if any proper 2-coloring g of H satisfies

$$(\forall x \in V) f(x) = g(x) \vee (\forall x \in V)(f(x) = g(x) + 1 \vee f(x) + 1 = g(x))$$

In this case, we call H *uniquely properly 2-colorable.*

Theorem 4.1.2. *Over* RCA_0*, the following are equivalent*

1. ATR_0
2. *If* $\langle H_i \rangle_{i\in\mathbb{N}}$ *is a sequence of hypergraph each of which admits at most one proper 2-coloring then there is a function* $f : \mathbb{N} \to 2$ *such that* $f(i) = 1$ *if and only if* H_i *has a proper 2-coloring.*

Showing that ATR_0 suffices to prove (2) amounts to observing that being a proper 2-coloring of a hypergraph H is an arithmetic (over H) property of sets. The reversal utilizes the formulation of ATR_0 in terms of finding what we call *uniquely ill-founded* trees in a given sequence $\langle T_i \rangle_{i\in\omega}$ of subtrees of $\omega^{<\omega}$. We say a tree T is *ill-founded* if it is not well-founded. We say a tree T is *uniquely ill-founded* if it has exactly one path.

For each tree T_i, we construct a hypergraph H_i via the the methods of Davis, Hirst, Pardo, and Ransom [3]. This tree T_i will be uniquely ill-founded if and only if H_i is uniquely properly 2-colorable. We require *leaf sets* for the construction.

Definition 4.1.3. For a tree $T \subseteq \omega^{<\omega}$ the set $L_T = \{\sigma : \forall n(\sigma * n \notin T)\}$ is the *leaf set* of T.

For a detailed treatment of trees and their leaf sets in reverse mathematics, see Hirst [17]. We summarize the key facts.

Naively, L_T is Π^0_1-definable in T. This is in fact optimal for arbitrary trees as Hirst [17] showed that over RCA_0 finding the leaf set for a given tree is equivalent to ACA_0. However, for certain trees this process is computable in T and thus can be borne out in RCA_0. The following appears in Lemma 5 and Theorem 6 of [17].

Proposition 4.1.4 (RCA_0)**.** *If* $T \subseteq \omega^{<\omega}$ *is a tree then*

$$T^+ = \big\{\tau : (\exists \sigma \in T)\big(|\sigma| = |\tau| \land (\forall n < |\sigma|)(\tau(n) = \sigma(n) + 1)\big)\big\}$$

and $T^* = T^+ \cup \{\tau * 0 : \tau \in T^+\}$ *are also trees and* L_{T^*} *exists. Moreover,* T *is well-founded if and only if* T^* *is well-founded.*

Proof. Clearly, T^+ is computable in T and T^* is computable in T so these sets exists. Since T is closed under initial segments, so is T^+ and in turn, so is T^*.

Note $\sigma \in L_{T^*}$ if and only if $\sigma \in T^*$ and $\sigma(|\sigma| - 1) = 0$. So L_{T^*} is also computable in T^*, and thus T.

Finally, f is a path in T if and only if g is a path in T^* where for all n, $g(n) = f(n) + 1$. □

Corollary 4.1.5. *Given a sequence of trees* $\langle T_i \rangle_{i \in \omega}$*, we may form in* RCA_0 *the sequence* $\langle T_i^*, L_{T_i} \rangle_{i \in \omega}$.

This conversion renders additional information in certain principles equivalent to ATR_0 and Π^1_1-CA_0. For instance, Hirst showed the following among other equivalences in Theorem 7 of [17].

Lemma 4.1.6. *Over* RCA_0*, the following are equivalent:*

1. ATR_0
2. *If* $\langle T_i \rangle_{i \in \mathbb{N}}$ *is a sequence of trees each with at most one infinite path, then there is a set* Z *such that for all* i*,* $i \in Z$ *if and only if* T_i *has an infinite path.*
3. *If* $\langle T_i, L_i \rangle_{i \in \mathbb{N}}$ *is a sequence of trees each with at most one infinite path and* L_i *is the leaf set of* T_i*, then there is a set* Z *such that for all* i*,* $i \in Z$ *if and only if* T_i *has an infinite path.*

Proof. This is immediate from Theorem V.5.5 of Simpson [22] combined with Corollary 4.1.5. □

We are now ready to give the construction of a hypergraph H from a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$. We follow the procedure of Davis, Hirst, Pardo, and Ransom [3], see Theorem 6.

Lemma 4.1.7 (RCA_0)**.** *Given a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$ with leaf set L, there exists a hypergraph H such that H is properly 2-colorable if and only if T is ill-founded.*

Proof. We construct H uniformly from T and L as follows: The vertices of H are $\{a_0, a_1, b_0, b_1, s\}$ together with vertices labeled σ_0 and σ_1 for each nonempty sequence $\sigma \in T$. The edges of H consist of

- $(a_0, a_1), (a_1, s), (b_0, b_1)$ and (b_1, s),
- (σ_0, σ_1) for every nonempty $\sigma \in T$,
- (σ_1, s) if σ is a leaf of T,
- $E_\sigma = \{\sigma_1\} \cup \{\tau_0 : \tau \in T \wedge \tau^{\#} = \sigma\}$ if $\sigma \in T$ is not a leaf, and
- $E_0 = \{a_0, b_0\} \cup \{\sigma_0 : \sigma \in T \wedge |\sigma| = 1\}$.

Note RCA_0 proves H exists as V and E are computable in T and L.

For the if direction, suppose T has an infinite path f. Denote $f \restriction n$ by f^n. Define c, a proper 2-coloring of H, as follows. Let

$$\begin{aligned} c(s) = c(a_0) = c(b_0) = 1, \quad c(a_1) = c(b_1) = 0, \\ c(f^i_0) = 0, \quad c(f^i_1) = 1, \text{ for all } i \in \mathbb{N}, \\ c(\tau_0) = 1 \text{ and } c(\tau_1) = 0 \text{ for } \tau \not\prec f. \end{aligned}$$

It is simple to verify by cases that c is a proper 2-coloring: argue individually for each type of edge in H.

For the only if direction, suppose c is a proper 2-coloring of the vertices in H. Without loss of generality, assume $c(s) = 0$. Then by the first bullet, $c(a_1) = c(b_1) = 1$ and $c(a_0) = c(b_0) = 0$. By the second bullet, for each $\sigma \in T$, $c(\sigma_0) = c(\sigma_1) \pm 1$. Combining this with the third bullet yields that $\sigma \in T$ is a leaf only if $c(\sigma_0) = 0$. As a_0 and b_0 have color 0, the fifth bullet guarantees a $\sigma \in T$ of length 1 with $c(\sigma_0) = 1$. Fix this σ and note it is not a leaf of T. As $c(\sigma_1) = 0$, there must be some immediate successor τ of σ with $c(\tau_0) = 1$ to properly color E_σ. Thus τ is not a leaf in T, and we may repeat this process to define a infinite path in T. $\square$

We are now ready to prove Theorem 4.1.2.

Proof. Suppose we are given a sequence of hypergraphs $\langle H_i \rangle_{i \in \mathbb{N}}$ each of which admits at most one proper 2-coloring. Working in ATR_0, we define a set Z such that $i \in Z$ if and only if H_i admits a proper 2-coloring. By Theorem V.5.2 of Simpson [22] ATR_0 proves the scheme

$$\forall i (\exists \text{ at most one } X) \varphi(i, X) \to \exists Z \forall i (i \in Z \leftrightarrow \exists X \varphi(i, X)),$$

where $\varphi(i, X)$ is any arithmetical formula in which Z does not occur. Note that X encoding a proper 2-coloring is arithmetic in H since we can effectively check each $e \in E$. To see this, consider a map $f : V \to 2$ for a hypergraph $H = \langle V, E \rangle$. Then f is a proper 2-coloring only if

$$(\forall u \in V) \forall n \, (u \in e_n \to (\exists v \in e_n)(f(u) \neq f(v))) \, .$$

Thus the formula $\varphi(i, X)$ which says X encodes a proper 2-coloring (and the coloring obtained by flipping the outputs) of H_i is arithmetic. By assumption each H_i can admit at most one such X, so ATR_0 proves the existence of a set Z such that $i \in Z$ if and only if H_i has a proper 2-coloring.

For the reversal let $\langle T_i, L_i \rangle_{i\in\mathbb{N}}$ be a sequence of trees in $\mathbb{N}^{<\mathbb{N}}$ equipped with leaf sets. By Lemma 4.1.6 it suffices to use item 2 to define the characteristic function of the set of indices of the ill-founded trees in $\langle T_i, L_i \rangle_{i\in\mathbb{N}}$.

Given the sequence $\langle T_i, L_i \rangle_{i\in\mathbb{N}}$, we may construct a sequence $\langle H_i \rangle$ of hypergraphs as in Lemma 4.1.7 as this procedure is uniform in the tree and leaf set. Moreover, each H_i has a proper 2-coloring if and only if T has an infinite path and the set of edges naturally is given as a sequence by lexicographic ordering on strings in T_i. Recall the vertices of H_i are $\{a_0, a_1, b_0, b_1, s\}$ together with vertices labeled σ_0 and σ_1 for each nonempty sequence $\sigma \in T_i$. The edges of H_i consist of

- $(a_0, a_1), (a_1, s), (b_0, b_1)$ and (b_1, s),
- (σ_0, σ_1) for every nonempty $\sigma \in T_i$,
- (σ_1, s) if σ is a leaf of T_i,
- $E_\sigma = \{\sigma_1\} \cup \{\tau_0 : \tau \in T_i \wedge \tau^{\#} = \sigma\}$ if $\sigma \in T_i$ is not a leaf, and
- $E_0 = \{a_0, b_0\} \cup \{\sigma_0 : \sigma \in T_i \wedge |\sigma| = 1\}$.

Claim. *For each $i \in \mathbb{N}$, T_i has exactly one infinite path if and only if H_i has a unique proper 2-coloring.*

To prove the claim, fix i and let $T = T_i$ and $H = H_i$. For the forward direction, suppose T has a unique infinite path. Then H has a proper 2-coloring $c : \mathbb{N} \to 2$. To show that c is unique, suppose d were a distinct proper 2-coloring of the vertices of H.

Then c and d satisfy

$$\neg(\forall v \in H(c(v) = d(v))) \wedge \neg(\forall v \in H(c(v) \neq d(v))).$$

Assume $c(s) = d(s) = 1$ by flipping the colors if necessary.

We show by induction that $c = d$. Note $c(a_1) = c(b_1) = 0$, and so $c(a_0) = c(b_0) = 1$. To properly color E_0, both c and d must assign σ_0 color 1 for possibly distinct length one 1 strings $\sigma \in T$. As T has a unique path, we claim there can only be one such σ. Suppose not: then, the proof of Lemma 4.1.7 allows us to extend both distinct witnessing strings for c and d to two infinite paths in T, a contradiction. Suppose inductively that $c(\sigma_0) = d(\sigma_0)$ and $c(\sigma_1) = d(\sigma_1)$ for all nodes σ of length less than or equal to n, and fix τ such that $|\tau| \leq n$, and $c(\tau_0) = d(\tau_0) = 1$. Consider an arbitrary string $\sigma \in T$ of length $n + 1$. If σ is not extendible to an infinite path in T, then both c and d must assign σ_0 color 1 and and σ_1 color 0. If σ is extendible, it is the only such string in T of length $n + 1$. By induction, τ is extendible, so σ is a successor of τ. Consequently, all other successors of τ are not extendible, and we have $c(\sigma_0) = d(\sigma_0) = 0$ as E_τ is properly 2-colored. This completes the induction.

For the reverse direction, we prove the contrapositive: if T has two paths, then H has more than one proper 2-coloring (up to swapping colors). Let f and g be two distinct paths in T. Let f^i denote $\langle f(0), f(1), \ldots, f(i)\rangle$ and g^i denote $\langle g(0), g(1), \ldots, g(i)\rangle$. Define the proper 2-colorings c_f and c_g of H as in the proof of

Lemma 4.1.7:

$$c_f(s) = c_f(a_0) = c_f(b_0) = 1, \quad c_f(a_1) = c_f(b_1) = 0,$$
$$c_f(f_0^i) = 0, \quad c_f(f_1^i) = 1, \text{ for all } i \in \mathbb{N},$$
$$c_f(\tau_0) = 1 \text{ and } c_f(\tau_1) = 0 \text{ for } \tau \not\prec f;$$

$$c_g(s) = c_g(a_0) = c_g(b_0) = 1, \quad c_g(a_1) = c_g(b_1) = 0,$$
$$c_g(g_0^i) = 0, \quad c_g(g_1^i) = 1, \text{ for all } i \in \mathbb{N},$$
$$c_g(\tau_0) = 1 \text{ and } c_g(\tau_1) = 0 \text{ for } \tau \not\prec g.$$

Note $c_f(s) = c_g(s) = 1$. Fix n such that $f^n \neq g^n$. For $m > n$, we have $c_f(g_1^m) = 0$ as $g^m \not\prec f$. Since $c_g(g_1^m) = 1$, we see c_f and c_g are distinct proper 2-colorings of H. This verifies the claim.

To complete the proof, we define the characteristic function of the set of indices for the ill-founded trees in $\langle T_i, L_i \rangle$. Apply item 2 to determine a function $f : \mathbb{N} \to 2$ such that $f(i) = 1$ if and only if H_i has a proper 2-coloring. Then by Δ_1^0 comprehension the set $Z = \{i : f(i) = 1\}$ exists. Since T_i has a path if and only if H_i has a proper 2-coloring if and only if $f(i) = 1$, the set Z is as desired. $\square$

It is natural to state the principles at play in Theorem 4.1.2 as *problems* in the sense of Weihrauch reductions. Let $\mathsf{UHPC}(2)$ be the problem whose instances are hypergraphs H that admit at most one distinct proper 2-coloring, with solution 1 if H does admit a proper 2-coloring and 0 if not. Similarly define UIF to be the problem which has for instances trees $T \subseteq \mathbb{N}^{\mathbb{N}}$ that contain as most one path with the solution 1 if T has exactly one infinite path, and 0 otherwise. Thus UIF is the problem of determining whether or not a tree is ill-founded given that it will be uniquely ill-founded if so.

Hirst formulated similar problems for the colorability of general hypergraphs in Theorem 12 of [17]. Specifically, he defined the problems $\mathsf{HPC}(k)$: instances are hypergraphs H with edge set E given as a sequence $\langle e_i \rangle_{i \in \mathbb{N}}$, and solutions are the integer 1 if H admits a proper k-coloring and the integer 0 if not; and WF: instances are trees $T \subseteq \mathbb{N}^{\mathbb{N}}$ with solutions the integer 1 if T is well-founded and 0 otherwise. The infinite parallelization of WF, written $\widehat{\mathsf{WF}}$ is a natural analogue of the reverse mathematical system $\Pi^1_1\text{-}\mathsf{CA}_0$ in the Weihrauch degrees. The search for a definitive analogue of the system ATR_0 in the Weihrauch lattice is of current interest. For a survey of potential candidates, see Kihara, Marcone and Pauly [20]. Note $\widehat{\mathsf{WF}}$ is denoted by $\Pi^1_1\text{-}\mathsf{CA}_0$ in [20], while we reserve the expression $\Pi^1_1\text{-}\mathsf{CA}_0$ for the subsystem of second order arithmetic.

We modify Hirst's proof of Theorem 12 in [17] to show that $\mathsf{UHPC}(2)$ and UIF are strongly Weihrauch equivalent. This, in combination with Theorem 4.1.2, yields UIF as another analogue of ATR_0 in the strong Weihrauch lattice.

Theorem 4.1.8. $\mathsf{UHPC}(2) \equiv_{\mathsf{sW}} \mathsf{UIF}$.

Proof. To show that $\mathsf{UIF} \leq_{\mathsf{sW}} \mathsf{UHPC}(2)$, apply the reversal from Theorem 4.1.2. Given a tree T we may uniformly compute the tree T^* and its leaf set L_{T^*} as in Lemma 4.1.4. In turn, we may compute the graph H from Lemma 4.1.7 for T^*. Applying $\mathsf{UHPC}(2)$ yields a solution which is identically a solution of T. This is because H will have a proper 2-coloring if and only if T^*, and thus T, has an infinite path.

To see that $\mathsf{UHPC}(2) \leq_{\mathsf{sW}} \mathsf{UIF}$, let $H = \langle V, E \rangle$ be a hypergraph with vertices $V = \{v_0, v_1, \dots\}$ and edges $E = \{e_0, e_1, \dots\}$. Build a tree T uniformly in H as follows:

place σ in T if σ satisfies the following.

1. If $\sigma(2j) = k$, then either k is a (code for a) pair $\{v_m, v_n\} \subseteq e_j$ and m is the least indexed element of e_j, or k is (a code for) $\emptyset$ and e_j does not contain two vertices from $\{v_0, v_1, \dots, v_{|\sigma|}\}$.

2. The even entries are a 2-coloring: if $2j+1 < |\sigma|$, then $\sigma(2j+1) < 2$ which we view as the of v_j.

3. The partial coloring defined on the odd entries of σ is proper with respect to the even entries: if $\sigma(2j)$ encodes the set $\{v_m, v_n\}$ and $2m+1, 2n+1 < |\sigma|$ then $\sigma(2m+1) \neq \sigma(2n+1)$.

4. If $\sigma(2j)$ encodes the set $\{v_m, v_n\}$ with $m < n$ and $2n+1 < |\sigma|$, then the odd entries of σ are constant on all vertices in $v_i \in e_j$ with $i < n$.

5. If $|\sigma| > 1$, then $\sigma(1) = 0$.

Note T is closed under prefix and is thus a tree. We claim that H has a unique proper 2-coloring if and only if T has exactly one path. To show this, we first prove the contrapositive of the only if direction before directly showing the if direction.

To begin, if T does not have exactly one path, then T either has no path or has at two least paths. Any proper 2-coloring of H can be used to define a path through T, so if T has no path, H is not properly 2-colorable. On the other hand, suppose T has two distinct paths f and g and let c_f and c_g be the proper 2-colorings of H defined by their odd entries. Note $c_f(v_0) = c_g(v_0) = 0$ by item 5. To see c_f and c_g are distinct, we show f and g differ on some odd entry. Suppose not. So $c_f = c_g$. Note that the pairs encoded in the even entries of f and g witness that c_f and c_g are proper (items 1 and 3). Moreover, they witness this with the unique vertex of least index from each edge of cardinality at least 2 which differs in color from the least indexed vertex of the edge (item 4). Since f and g define the same coloring, they must agree on each of these pairs witnessing it is proper. Hence f and g agree on every even entry and are not distinct, a contradiction. Fix j such that $f(2j+1) \neq g(2j+1)$. Then $c_f(v_j) \neq c_g(v_j)$ and we have that c_f and c_g are distinct proper 2-colorings of H. In both cases we conclude that if T does not have exactly one path, then H does not have a unique proper 2-coloring. This verifies the only if direction of the claim.

For the if direction, suppose T has exactly one path. Then clearly H has a unique proper 2-coloring. If H had any more proper 2-colorings, one of them would be distinct from that given by the path of f while agreeing that the color of v_0 is 0. This coloring would define a second path in T diverging at the least odd entry in which the colors differ. This completes the verification of the claim and the proof is complete. □

4.2 Distinguishing colorability and unique colorability in graphs and hypergraphs — Joint work with Jeffry L. Hirst

The results of the previous section motivate the general program, being conducted by the author in collaboration with Jeff Hirst, of studying unique colorability in graphs and hypergraphs. We now give some preliminary results of this work.

As mentioned above, Theorem 6 of Davis, Hirst, Pardo, and Ransom [3] establishes Π^1_1-CA_0 is equivalent to determining which in a sequence of hypergraphs are properly 2-colorable. Specifically, they showed that over RCA_0, Π^1_1-CA_0 is equivalent to the following statement

If $\langle H_i \rangle_{i \in \mathbb{N}}$ is a sequence of hypergraphs, then there is a function $f : \mathbb{N} \to 2$ such that $f(i) = 1$ if and only if H_i has a proper k-coloring.

This can be extended to find a function which discriminates between hypergraphs with a unique proper 2-coloring and those without.

Theorem 4.2.1. *Over RCA_0, the following are equivalent:*

1. Π^1_1-CA_0

2. *If $\langle H_i \rangle_{i \in \mathbb{N}}$ is a sequence of hypergraphs, then there is a function $s : \mathbb{N} \to 3$ such that*

$$s(i) = \begin{cases} 0 & \textit{if } H_i \textit{ has no proper 2-coloring} \\ 1 & \textit{if } H_i \textit{ has a unique proper 2-coloring} \\ 2 & \textit{if } H_i \textit{ has many proper 2-colorings} \end{cases}$$

Proof. Suppose we are given the sequence $\langle H_i \rangle_{i \in \mathbb{N}}$ and work in Π^1_1-CA_0. As mentioned above, Π^1_1-CA_0 suffices to obtain a function $t : \mathbb{N} \to 2$ such that $t(i) = 1$ if and only if H_i admits a proper 2-coloring. Let $\varphi(i, X)$ be the arithmetic formula which says X encodes a proper 2-coloring of H_i. The set

$$S = \{i : t(i) = 1 \wedge \forall f \forall g (\varphi(i, f) \wedge \varphi(i, g)) \to (\forall n f(n) = g(n) \vee \forall n f(n) = g(n) + 1)\}$$

is Π^1_1 definable. Define $s : \mathbb{N} \to 3$ as follows:

$$s(i) = \begin{cases} 0 & \text{if } t(i) = 0 \\ 1 & \text{if } i \in S \\ 2 & \text{else} \end{cases}$$

Clearly s is definable from S and t and satisfies the consequent of item (2). This completes the direction that 1 implies 2.

The direction 2 implies 1 is immediate from Theorem 6 of Davis, Hirst, Pardo, and Ransom [3]. $\square$

We next restrict our attention to graphs. Note that k-colorability of a graph $G = (V, E)$ is a compactness phenomenon. That is, a countable graph G is k-colorable if and only if every finite subgraph of G is k-colorable. Indeed, with an enumeration of $V = \langle v_i \rangle_{i \in \mathbb{N}}$, it is simple to define the tree $T \subseteq k^{<\mathbb{N}}$ of k-colorings of initial segments $(\{v_i : i \leq n\}, E)$ of G. Place $\sigma \in T$ if $|\sigma| = n$, and $c(v_i) = \sigma(i)$ is a k-coloring of $(\{v_i : i \leq n\}, E)$. Any path in T encodes a k-coloring of G, and if T is well-founded, there is some m such that the subgraph $(\{v_i : i \leq m\}, E)$ is not k-colorable. See Theorems 3.12 and 3.13 of Hirst [14] for a proof that this fact is equivalent to WKL_0.

So we see that determining a given graph is not k-colorable is a Σ^0_1 question. We need merely determine if there exists a (code for a) finite subgraph of G which is not k-colorable, and there are only finitely many possible k-colorings to check. Determining *unique* k-colorability however, is a more complex question. We begin with 2 colors.

Definition 4.2.2. A graph $G = (V, E)$ is called *connected* if for every pair of vertices u, v, there is a sequence $\langle v_0, \ldots, v_n \rangle$ of vertices with $v_0 = u$, $v_n = v$, and $\{v_i, v_{i+1}\} \in E$ for all $i \leq n$. We call $\langle v_0, \ldots, v_n \rangle$ a path from u to v.

If a 2-colorable graph is connected, it may only admit a unique 2-coloring. Any change in the color of a vertex will propagate to every other vertex via some path. For example, if c and d were distinct 2-colorings of a connected graph, there would be two vertices u and v such that $c(u) = d(u)$ and $c(v) \neq d(v)$. By working on a path, we can assume without loss of generality that u and v are adjacent. Since c and d are 2-colorings, one of c or d assigns the same color to u and v, a contradiction. As determining whether a graph is connected is also arithmetic, we may distinguish between 2-colorable and uniquely 2-colorable graphs in ACA_0.

Theorem 4.2.3. *Over* RCA_0*, the following are equivalent:*

1. ACA_0

2. *If* $\langle G_i \rangle_{i\in\mathbb{N}}$ *is a sequence of graphs, then there is a function* $s : \mathbb{N} \to 3$ *such that*

$$s(i) = \begin{cases} 0 & \textit{if } G_i \textit{ has no 2-coloring} \\ 1 & \textit{if } G_i \textit{ has a unique 2-coloring} \\ 2 & \textit{if } G_i \textit{ has many 2-colorings} \end{cases}$$

Proof. To see 1 implies 2, note that the value of $s(i)$ is arithmetic in $G_i = (V_i, E_i)$. To elaborate, G_i has no 2-coloring if some finite subgraph of G_i is not 2-colorable. So $s(i) = 0$ if and only if there exists (a code for) a finite subset of V_i, such that for all (codes for) functions $c : F \to 2$, there are $u, v \in F$ such that $(u, v) \in E_i$ and $c(u) = c(v)$. Note $s(i) \neq 0$ is also arithmetic in G_i.

Specifically, $s(i) = 1$ if both $s(i) \neq 0$ and G_i is connected. That is $s(i) = 1$ if and only if both $\neg(s(i) = 0)$ and for every pair $u, v \in V_i$ there exists (a code for) a path $\langle v_0, \ldots, v_n \rangle$ from u to v. We have $s(i) = 2$ in case both $s(i) \neq 0$ and $s(i) \neq 1$.

Thus arithmetic comprehension suffices to prove s exists.

For the reversal, we take an arbitrary injection $f : \mathbb{N} \to \mathbb{N}$ and show $\mathrm{ran}(f)$ exists, which suffices by Lemma III.1.3 of Simpson [22]. Construct a sequence of graphs

$\langle G_i \rangle_{i \in \mathbb{N}}$ as follows. For each n, let $G_n = (\mathbb{N}, E_n)$ with $E_n = \{(m, m+1) : f(m) \neq n\}$. Recursive comprehension suffices to show the sequence $\langle G_i \rangle_{i \in \mathbb{N}}$ exists.

Apply item 2 to $\langle G_i \rangle_{i \in \mathbb{N}}$ to obtain s. Note if $f(m) = n$, then G_n is not connected. Indeed, since f is injective, $\{0, \dots m\}$ and $\{m+1, m+2, \dots\}$ are disjoint connected components of G_n. If $n \notin \operatorname{ran}(f)$, then G_n is connected, and clearly 2-colorable. Hence, $n \in \operatorname{ran}(f)$ if and only if $s(n) \neq 1$. So $\operatorname{ran}(f)$ exists by recursive comprehension, and the proof is complete. $\square$

To extend this to k-colorings for $k > 2$, we invoke another compactness argument. Namely, that G has more than one k-coloring only if it has a finite subgraph G' with more than one k-coloring, each of which can be extended to any finite supergraph of G'. If this is the case, we can construct trees in $k^{<\mathbb{N}}$ similar to above containing all finite extensions of some k-coloring of G' to finite supergraphs. These trees will necessarily be infinite and thus contain infinite paths each of which define a distinct k-coloring on G. However, as Theorem 4.2.5 shows, we require arithmetic comprehension to sort which graphs in a given sequence have one or many k-colorings. This is in contrast to Theorem 3.13 of Hirst [14].

Definition 4.2.4. We say $G = (V, E)$ is *uniquely k-colorable* if there exists a k-coloring $c : V \to k$ and for any other k-coloring d of G, there is a permutation σ on $\{0, \dots, k-1\}$ such that for all $v \in V$

$$d(v) = \sigma(c(v)).$$

Note distinct k-colorings c and d must both agree on some vertex and differ on another.

Theorem 4.2.5. *For each $k > 2$, the following are equivalent over* RCA_0*:*

1. ACA_0

2. *If $\langle G_i \rangle_{i \in \mathbb{N}}$ is a sequence of graphs, then there is a function $s : \mathbb{N} \to \{0, 1, 2\}$ such that*

$$s(i) = \begin{cases} 0 & \text{if } G_i \text{ has no } k\text{-coloring} \\ 1 & \text{if } G_i \text{ has a unique } k\text{-coloring} \\ 2 & \text{if } G_i \text{ has many } k\text{-colorings} \end{cases}$$

Proof. As in Theorem 4.2.3, note $s(i)$ is arithmetic in G_i. We have that $s(i) = 0$ if there is a (code for a) finite subgraph of G_i which has no k-coloring. We have that $s(i) = 2$ if there is a (code for a) finite subgraph G' of G_i with more than one k-coloring, and for every (code for a) finite supergraph of G', there is are k-colorings extending each of those on G'. Then $s(i) = 1$ if and only if $s(i) \neq 0$ and $s(i) \neq 2$. As s is arithmetically definable in $\langle G_i \rangle_{i \in \mathbb{N}}$, arithmetic comprehension suffices to prove it exists.

For the reversal, we again take an arbitrary injection $f : \mathbb{N} \to \mathbb{N}$ and show $\operatorname{ran}(f)$ exists. Construct a computable sequence of graphs $\langle G_i \rangle_{i \in \mathbb{N}}$ as follows. Let $G_n = (\mathbb{N} \times \{0, \ldots, k\}, E_n)$ and construct E_n in stages. At stage s, if $f(s) = n$, add edges to E_n to make $\langle s, 0 \rangle, \langle s, 1 \rangle, \ldots, \langle s, k \rangle$, a complete graph on $k + 1$ many vertices. Otherwise, do nothing.

Apply item 2 to $\langle G_i \rangle_{i \in \mathbb{N}}$ to obtain s. If $f(m) = n$, then G_n contains a subgraph which is a complete graph on $k + 1$ vertices. So G_n is not k-colorable. If $n \notin \operatorname{ran}(f)$, then G_n has a trivial edge relation, and is thus k-colorable. Hence, $n \in \operatorname{ran}(f)$ if and only if $s(i) = 0$. We see $\operatorname{ran}(f)$ exists by recursive comprehension, and the proof is complete. $\square$

Many questions remain in this direction and in particular in the analysis of these results under computable and Weihrauch reductions. For instance, we conjecture item 2 of Theorem 4.2.1 is strongly Weihrauch equivalent to $\widehat{\mathsf{WF}}$. Item 2 in Theorem 4.2.5 seems to indicate the number of colors is innocuous when it comes to sorting graphs by unique k-colorability. We seek to understand whether the differences in sorting for say 2 or 5-colorability could be made precise using computable or Weihrauch reductions.

Purely combinatorial questions remain as well. Recall a strong k-coloring on a hypergraph is one that is injective on every edge. As we saw in the proof of Theorem 4.2.5, for each k, there is a graph with no k-coloring. We conjecture this holds as well for hypergraphs with respect to strong k-colorings.

Bibliography

[1] Eric P. Astor, Damir D. Dzhafarov, Reed Solomon, and Jacob Suggs, *The uniform content of partial and linear orders*, Annals of Pure and Applied Logic **168** (2017), no. 6, 1153–1171.

[2] Vasco Brattka, Guido Gherardi, and Arno Pauly, *Weihrauch complexity in computable analysis*, to appear, `arXiv:1707.03202`.

[3] Caleb Davis, Jeffry Hirst, Jake Pardo, and Tim Ransom, *Reverse mathematics and colorings of hypergraphs*, Archive for Mathematical Logic **58** (2019), no. 5-6, 575–585.

[4] Reinhard Diestel, *Graph theory*, third ed., Graduate Texts in Mathematics, vol. 173, Springer-Verlag, Berlin, 2005.

[5] Rodney G. Downey, Denis R. Hirschfeldt, Steffen Lempp, and Reed Solomon, *Computability-theoretic and proof-theoretic aspects of partial and linear orderings*, Israel Journal of Mathematics **138** (2003), 271–289.

[6] Damir D. Dzhafarov, *Strong reductions between combinatorial principles*, The Journal of Symbolic Logic **81** (2016), no. 4, 1405–1431.

[7] Damir D. Dzhafarov, Ludovic Patey, Reed Solomon, and Linda Brown Westrick, *Ramsey's theorem for singletons and strong computable reducibility*, Proceedings of the American Mathematical Society **145** (2017), no. 3, 1343–1355.

[8] Marshall Hall, Jr., *Distinct representatives of subsets*, Bulletin of the American Mathematical Society **54** (1948), 922–926.

[9] Philip Hall, *On representatives of subsets*, J. London Math. Soc. **10**, 26–30, DOI 10.1112/jlms/s1-10.37.26.

[10] E. Herrmann, *Infinite chains and antichains in computable partial orderings*, The Journal of Symbolic Logic **66** (2001), no. 2, 923–934.

[11] Denis R. Hirschfeldt, *Slicing the Truth: On the Computable and Reverse Mathematics of Combinatorial Principles*, Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore, vol. 28, World Scientific, Singapore, 2015.

[12] Denis R. Hirschfeldt and Carl G. Jockusch, Jr., *On notions of computability-theoretic reduction between Π^1_2 principles*, Journal of Mathematical Logic **16** (2016), no. 1, 1650002, 59.

[13] Denis R. Hirschfeldt and Richard A. Shore, *Combinatorial principles weaker than Ramsey's theorem for pairs*, The Journal of Symbolic Logic **72** (2007), no. 1, 171–206.

[14] Jeffry L. Hirst, *Combinatorics in subsystems of second order arithmetic*, Ph.D. thesis, The Pennsylvania State University, 1987.

[15] Jeffry L. Hirst, *Marriage theorems and reverse mathematics*, Logic and computation (Pittsburgh, PA, 1987), Contemp. Math., vol. 106, Amer. Math. Soc., Providence, RI, 1990, pp. 181–196.

[16] ______, *Representations of reals in reverse mathematics*, Bulletin of the Polish Academy of Sciences, Mathematics **55** (2007), no. 4, 303–316.

[17] ______, *Leaf management*, Computability. The Journal of the Association CiE **9** (2020), no. 3-4, 309–314.

[18] Jeffry L. Hirst and Noah A. Hughes, *Reverse mathematics and marriage problems with unique solutions*, Archive for Mathematical Logic **54** (2015), no. 1-2, 49–57.

[19] ______, *Reverse mathematics and marriage problems with finitely many solutions*, Archive for Mathematical Logic **55** (2016), no. 7-8, 1015–1024.

[20] Takayuki Kihara, Alberto Marcone, and Arno Pauly, *Searching for an analogue of* ATR_0 *in the Weihrauch lattice*, The Journal of Symbolic Logic, to appear, `arXiv:1812.01549`.

[21] David Nichols, *Effective techniques in reverse mathematics*, Ph.D. thesis, University of Connecticut, 2019.

[22] Stephen G. Simpson, *Subsystems of second order arithmetic*, second ed., Perspectives in Logic, Cambridge University Press, Cambridge; Association for Symbolic Logic, Poughkeepsie, NY, 2009.

[23] Robert I. Soare, *Turing computability*, Theory and Applications of Computability, Springer-Verlag, Berlin, 2016.

[24] Henry Towsner, *Constructing sequences one step at a time*, Journal of Mathematical Logic **20** (2020), no. 3, 2050017, 43.

[25] A. M. Turing, *On Computable Numbers, with an Application to the Entscheidungsproblem*, Proceedings of the London Mathematical Society. Second Series **42** (1936), no. 3, 230–265.

[26] Rebecca Weber, *Computability theory*, Student Mathematical Library, vol. 62, American Mathematical Society, Providence, RI, 2012.