MATH 873 F19 NOTES (IN PROGRESS)

1. Introduction

Slaman:

> [..] we are attempting to understand the interaction between the mathematical objects and the means needed to speak about them.

Themes of this course:

- Computability as a unifying and organizing principle spanning different fields (e.g., combinatorics, analysis, topology)
- Interactions between computable reducibilities and reverse mathematics

Examples of problems:

- Given a continuous function $f : \mathbb{R} \to \mathbb{R}$ with a zero, find a zero;
- Given a convergent sequence of real numbers, find its limit;
- Given a bounded sequence of real numbers, find an accumulation point;
- Given a sequence of natural numbers, find its minimum;
- Given a bounded sequence of natural numbers, find its maximum;
- Given a finitely branching infinite subtree of $\mathbb{N}^{<\mathbb{N}}$, find an infinite path;
- Given a countable open cover of $[0, 1]$, find a finite subcover;
- Given a continuous function $f : [0, 1] \to \mathbb{R}$ such that $f(0)$ and $f(1)$ have opposite signs, find a zero.

A basic question of computable mathematics:

> Can we solve these problems computably? If not, can we measure how noncomputable they are?

Some early (i.e., 1950's) results in the area:

- There is a computable bounded increasing sequence of real numbers whose limit is not computable (Specker);
- There is an infinite computable subtree of $2^{<\mathbb{N}}$ with no computable path (Kleene).

Date: December 2, 2019.

The above results imply that the corresponding problems cannot be solved computably, for any computable way to solve a problem certainly must produce a computable solution for each computable instance.

What about the other problems above? Can they be solved computably? How can we formalize this notion?

1.1. **Computability on $\mathbb{N}^\mathbb{N}$.** The first order of business is to develop a notion of computability on spaces other than $\mathbb{N}$. For any countable space S, such as the space of all finite strings in a finite alphabet, we can transfer notions of computability from $\mathbb{N}$ using a surjection $\nu :\subseteq \mathbb{N} \to S$ (known as a numbering). For example, $e \mapsto W_e$ is a (choice of) numbering.

What about spaces like $\mathbb{R}$? The space of continuous functions on $\mathbb{R}$? The space of (bounded) sequences in $\mathbb{R}$? For those, we can transfer notions of computability from $\mathbb{N}^\mathbb{N}$. So let us first define computability on $\mathbb{N}^\mathbb{N}$.

Definition 1.1. A function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ is *computable* if there is a total computable function $f : \mathbb{N}^{<\mathbb{N}} \to \mathbb{N}^{<\mathbb{N}}$ such that:

- if $\sigma \preceq \tau$, then $f(\sigma) \preceq f(\tau)$;
- $F(x) = y$ if and only if for all m, there exists n such that $f(x \upharpoonright n) \succeq y \upharpoonright m$.

This definition can be relativized to set oracles (i.e., F is A-computable if there is a total A-computable f.)

Alternatively, one can define computable functions using Type-2 Turing machines. Roughly speaking, these are Turing machines with one-way input and output tapes: given sufficiently long initial segments of a valid input, the machine outputs arbitrarily long initial segments of the appropriate element of $\mathbb{N}^\mathbb{N}$. See Weihrauch's book [56] for details.

The above notion of computability is an effective refinement of continuity. We make this precise as follows.

First, we define a topology on $\mathbb{N}^\mathbb{N}$ (Baire space) which we will use for the rest of the course. We use the product topology induced by the discrete topology on $\mathbb{N}$. For each $\sigma \in \mathbb{N}^{<\mathbb{N}}$, $[\sigma]$ is defined to be $\{x \in \mathbb{N}^\mathbb{N} : \sigma \prec x\}$. $\{[\sigma] : \sigma \in \mathbb{N}^{<\mathbb{N}}\}$ is a countable base for this topology. Each $[\sigma]$ is clopen. The open sets are exactly the $\boldsymbol{\Sigma^0_1}$ sets.

Theorem 1.2 (folklore?)**.** *If $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ is computable, then it is continuous with Π^0_2 domain. Conversely, if $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ is continuous with $\boldsymbol{\Pi^0_2}$ domain (i.e., G_δ domain), then it is computable relative to some oracle (in $\mathbb{N}^\mathbb{N}$).*

Proof. ($\Rightarrow$). The point is that any finite portion of the output depends only on a finite portion of the input.

Formally, it suffices to show that for each $\sigma \in \mathbb{N}^{<\mathbb{N}}$, the preimage of $\sigma^\frown\mathbb{N}^\mathbb{N}$ is relatively open in $\mathrm{dom}(F)$. Fix $f : \mathbb{N}^{<\mathbb{N}} \to \mathbb{N}^{<\mathbb{N}}$ which witnesses that F is computable. Suppose that $F(x) \in \sigma^\frown\mathbb{N}^\mathbb{N}$, i.e., $F(x)$ extends σ. Then there is some $m \in \mathbb{N}$ such that $f(x \upharpoonright m)$ extends σ. It follows that for any y extending $x \upharpoonright m$, if $F(y)$ is defined, then it extends $f(x \upharpoonright m)$. Hence $F^{-1}[\sigma^\frown\mathbb{N}^\mathbb{N}]$ contains $\mathrm{dom}(F) \cap ((x \upharpoonright m)^\frown\mathbb{N}^\mathbb{N})$.

The domain of F is Π^0_2 because

$$\mathrm{dom}(F) = \{x : \forall m \exists n[|f(x \upharpoonright n)| = m]\}.$$

($\Leftarrow$). Given some continuous $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$, consider the function $f : \mathbb{N}^{<\mathbb{N}} \to \mathbb{N}^{<\mathbb{N}}$ defined as follows: $f(\sigma)$ is the ρ of maximum length $< |\sigma|$ such that $F''[\mathrm{dom}(F) \cap [\sigma]] \subseteq [\rho]$ (in words: if x extends σ and $F(x)$ is defined, then $F(x)$ extends ρ). (If no such ρ exists, just define $f(\sigma) = \sigma$.)

Note that f can be encoded as an element of $\mathbb{N}^\mathbb{N}$.

Using f, we can compute some continuous $G :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ as follows: $G(x) = y$ if for all m, there exists n such that $f(x \upharpoonright n) = y \upharpoonright m$.

G always extends F, but the converse need not be true. (If it were true, that would be disturbing—we haven't used the assumption on $\mathrm{dom}(F)$!)

So we need to adjust things a little. This is where we make use of the assumption that $\mathrm{dom}(F)$ is $\mathbf{\Pi^0_2}$. Fix g such that

$$x \in \mathrm{dom}(F) \Leftrightarrow \forall m \exists \sigma[g(\sigma, m) = 1 \text{ and } x \in [\sigma]].$$

(Intuitively, we are fixing a sequence of open sets whose intersection is $\mathrm{dom}(F)$ and $g(\tau, m) = 1$ means that "the m^{th} open set contains $[\tau]$".)

Note that g can be encoded as an element of $\mathbb{N}^\mathbb{N}$.

We show that $f \oplus g$ computes F as follows. Vaguely speaking, we use g to slow f down. For each σ, we can compute (using g) the maximum $l(\sigma) \leq |\sigma|$ such that

$$\forall m < l(\sigma) \exists \tau \prec \sigma[g(\tau, m) = 1].$$

Then, define h as follows: for each σ, restrict the output of $f(\sigma)$ to length $l(\sigma)$ (if needed). We end by showing that $F(x) = y$ if and only if for all m, there exists n such that $h(x \upharpoonright n) = y \upharpoonright m$.

($\Rightarrow$) is clear, if you already believe that G extends F. As for ($\Leftarrow$), observe that the assumption implies that l is cofinal on initial segments of x. By definition of l, it follows that x lies in $\mathrm{dom}(F)$.

(Intuitively, if x does not lie in the m^{th} open set, then h will never output strings of length $> m$ when given initial segments of x.) $\square$

1.2. **Represented spaces and computability on them.**

Definition 1.3 (Kreitz, Weihrauch [44])**.** For any space X (of cardinality at most $\mathbb{N}^\mathbb{N}$), a *($\mathbb{N}^\mathbb{N}$)-representation of* X is a (possibly partial) surjection $\delta :\subseteq \mathbb{N}^\mathbb{N} \to X$. A *represented space* is a pair (X, δ), where δ is a representation of X.

Let (X, δ) be a represented space. If $\delta(p) = x$, then p is said to be a *(δ-)name of* x. If x has a computable δ-name, then it is said to be *(δ-)computable*.

Next, we define computability on functions between represented spaces.

Definition 1.4. Let (X, δ_X) and (Y, δ_Y) be represented spaces. Then $f :\subseteq X \to Y$ is *((δ_X, δ_Y)-)computable* (*(δ_X, δ_Y)-continuous* resp.) if there is some computable (continuous resp.) function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ such that

$$f(\delta_X(p)) = \delta_Y(F(p))$$

for every $p \in \mathrm{dom}(f \circ \delta_X)$. In this case, F is said to be a *realizer* of f.

$$\begin{array}{ccc} \mathbb{N}^\mathbb{N} & \xrightarrow{F} & \mathbb{N}^\mathbb{N} \\ {\scriptstyle \delta_X}\downarrow & & \downarrow{\scriptstyle \delta_Y} \\ X & \xrightarrow{f} & Y \end{array}$$

In other words, F is a realizer of f if for any $x \in \mathrm{dom}(f)$, F takes every δ_X-name for x to a δ_Y-name for $f(x)$.

Note. Whether F is a realizer of f only depends on $F \upharpoonright \mathrm{dom}(f \circ \delta_X)$. This means that $f :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ can be $(\mathrm{id}, \mathrm{id})$-computable (where $\mathrm{id} : \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ is the identity representation) but *not* computable! In fact, one can show that $f :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ is $(\mathrm{id}, \mathrm{id})$-computable if and only if it is a restriction of some computable function.

Proposition 1.5. *If $f :\subseteq X \to Y$ is (δ_X, δ_Y)-computable and $g :\subseteq Y \to Z$ is (δ_Y, δ_Z)-computable, then $g \circ f :\subseteq X \to Z$ is (δ_X, δ_Z)-computable.*

Different representations can carry different amounts of "information". For example, we could represent a real number using sequences of rational numbers converging to it, or sequences of rational numbers *rapidly* converging to it. Knowing the first few terms of a convergent sequence gives us no information about the limit, while knowing the first few terms of a rapidly convergent sequence gives us an approximation of the limit. This idea can be captured by the notion of reducibility between representations:

Definition 1.6. Let δ_1 and δ_2 be representations of X. We say that δ_1 *(computably) reduces* to δ_2, written $\delta_1 \leq \delta_2$, if $\mathrm{id}_X : X \to X$ is (δ_1, δ_2)-computable. We say that δ_1 and δ_2 are *(computably) equivalent*, written $\delta_1 \equiv \delta_2$, if $\delta_1 \leq \delta_2$ and $\delta_2 \leq \delta_1$.

Intuitively, if $\delta_1 \leq \delta_2$, then δ_1 provides *more* information than δ_2 (given a δ_1-name for x, we can "forget" information to obtain a δ_2-name for x).

Proposition 1.7. *The following hold:*

(1) *If $\delta^1_X \leq \delta^2_X$ and $x \in X$ is δ^1_X-computable, then it is also δ^2_X-computable.*
(2) *If $\delta^2_X \leq \delta^1_X$ and $\delta^1_Y \leq \delta^2_Y$ and $f :\subseteq X \to Y$ is (δ^1_X, δ^1_Y)-computable, then it is (δ^2_X, δ^2_Y)-computable.*

Let us compare the following representations of the real numbers up to reducibility:

(1) converging sequences of rational numbers;
(2) rapidly converging sequences of rational numbers (say, $(q_n)_n$ such that if $n, n' > m$, then $|q_n - q_{n'}| < 2^{-m}$);
(3) expansion in base 10;
(4) Dedekind cuts ($\{q \in \mathbb{Q} : q < x\} \oplus \{q \in \mathbb{Q} : q > x\}$);

One can show that (4) < (3) < (2) < (1).

(4) < (3) because the characteristic function of $\mathbb{Q}$ is computable under (4) but not (3).

(3) < (2) because $x \mapsto 3 \cdot x$ is computable under (2) but not (3) (try multiplying numbers around $0.\overline{3}$ by 3).

(2) < (1) because of Specker's result (which implies that there are reals with computable (1)-names but no computable (2)-names).

For later purposes, we introduce representations for the set of closed subsets and the set of open subsets of a computable metric space.

Definition 1.8. A metric space (X, d) is a *computable metric space* if there is a dense sequence $(q_n)_n$ such that $(m, n) \mapsto d(q_m, q_n)$ is computable.

If X is a computable metric space (with fixed choice of a dense sequence $(q_n)_n$ witnessing that), we represent the set of closed subsets of X *negatively*, as follows: p is a name for a closed set $A \subseteq X$ if p enumerates rational open balls (centered at q_n's) whose union is the complement of A.

This induces a *positive* representation of the open subsets of X, i.e., p is a name for an open set $U \subseteq X$ if p enumerates rational open balls whose union is A.

Example 1.9. The following are computable metric spaces with their usual metrics: finite spaces, $\mathbb{N}$, $\mathbb{R}$, $[0,1]$, $2^{\mathbb{N}}$, $\mathbb{N}^{\mathbb{N}}$.

Remark 1.10. Using the Sierpinski space, one can define representations of closed subsets of general represented spaces. See Pauly [49].

Finally, there are several natural constructions on represented spaces. These will be useful when we define operations on problems.

Definition 1.11. Define the following constructions on represented spaces:

- Product: $\delta_{X_0 \times X_1}(\langle p, q\rangle) = (\delta_{X_0}(p), \delta_{X_1}(q))$
- Coproduct: $\delta_{X_0 \sqcup X_1}(i^\frown p) = (i, \delta_{X_i}(p))$
- Finite parallelization: $\delta_{X^*}(n^\frown\langle p_0, \ldots, p_{n-1}\rangle) = (\delta_X(p_0), \ldots, \delta_X(p_{n-1}))$
- Parallelization: $\delta_{X^{\mathbb{N}}}(\langle p_0, p_1, \ldots\rangle) = (\delta_X(p_n))_n$
- Continuous functions: If p encodes some monotone $f : \mathbb{N}^{<\mathbb{N}} \to \mathbb{N}^{<\mathbb{N}}$, let η_p denote the continuous function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, as defined by $F(x) = y$ iff for all m, there exists n such that $f(x \upharpoonright n) = y \upharpoonright m$. If η_p realizes some total (δ_X, δ_Y)-continuous $f : X \to Y$, define $\delta_{\mathcal{C}(X,Y)}(p)$ to be f.

Remark 1.12. The represented spaces form a Cartesian closed category. For details, see Brattka [5].

1.3. **Problems.** We began the introduction with a list of problems. But what are problems, anyway?

Definition 1.13. A *problem* is a (possibly partial) multivalued function $f :\subseteq X \rightrightarrows Y$ between represented spaces. The *domain* of f is $\mathrm{dom}(f) = \{x \in X : f(x) \neq \emptyset\}$. An element of $\mathrm{dom}(f)$ is called an *instance* of f. For any $x \in \mathrm{dom}(f)$, an element of $f(x)$ is called an *f-solution to x*.

We use the term multivalued function instead of relation, because they have different composition operations: the composition $f \circ g$ of multivalued functions f and g has domain

$$\{x : \forall y \in g(x)[f(y) \neq \emptyset]\}$$

rather than

$$\{x : \exists y \in g(x)[f(y) \neq \emptyset]\}.$$

This restriction on the domain of $f \circ g$ implies that the composition of realizers for f and g is a realizer for $f \circ g$.

Any theorem of the form $\forall X[\Theta(X) \to \exists Y \Lambda(X,Y)]$ corresponds naturally to the multivalued function $X \mapsto \{Y : \Lambda(X,Y)\}$, with domain $\{X : \Theta(X)\}$. In particular, any Π^1_2 statement (such as those studied in reverse mathematics) has a corresponding problem. But our scope is more general than that; we do not require that Θ and Λ are arithmetical. We also do not require that X and Y are subsets of $\mathbb{N}$: they could be elements of any represented space.

Definition 1.14. We say that $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is a *realizer* of a multi-valued function $f :\subseteq X \rightrightarrows Y$ if

$$\delta_Y(F(p)) \in f(\delta_X(p))$$

for every $p \in \mathrm{dom}(f \circ \delta_X)$. In other words, given a name for some $x \in \mathrm{dom}(f)$, F outputs a name for some element of $f(x)$.

Note. Given two names for the same $x \in \mathrm{dom}(f)$, F is free to output two different names for the same element of $f(x)$, or even names for two different elements of $f(x)$!

Various notions can be transferred to problems via their realizers. For example:

Definition 1.15. A problem is *computable* if it has a computable realizer. A problem is *continuous* if it has a continuous realizer. A problem f is *pointwise computable* if every f-instance x has an x-computable f-solution.

Since computable single-valued functions preserve pointwise computability (given some input, their output is computable in the input), computability implies pointwise computability.

Example 1.16. Consider the "compactness" problem HB_0: given a sequence of open sets which cover $[0, 1]$, find a finite subcover. This problem is computable: since there is a finite subcover, we can simply enumerate open sets until we have enough to cover $[0, 1]$.

Concretely, given finitely many names for open sets, whether they cover $[0, 1]$ is a Σ^0_1 fact, so we will eventually see it happen.

Example 1.17. Consider the "contrapositive" HB_1 of the "compactness" problem: given a sequence of open sets which does not cover $[0, 1]$, find an element of $[0, 1]$ which is not covered. We show that HB_1 is not even pointwise computable. We define a computable list of open intervals which do not cover $[0, 1]$ but cover every computable real, as follows. Think of each index e as an enumeration of rationals $(q^e_i)_i$. For each e, once q^e_e is enumerated, we add the open interval centered at q^e_e with radius $2^{-(e+1)}$ to our list.

The total measure of the intervals we enumerate is $\leq \sum_e 2^{-(e+1)} = 1/2 < 1$, so the intervals do not cover $[0, 1]$.

Example 1.18. Consider the problem WKL: given an infinite subtree of $2^{<\mathbb{N}}$, produce an infinite path. (The fact that there is always an infinite path is known as weak König's lemma.)

WKL is not even pointwise computable, because there is an infinite computable subtree of $2^{<\mathbb{N}}$ with no computable path (Kleene).

Example 1.19. Consider the problem IVT corresponding to the intermediate value theorem: given a continuous function $f : [0,1] \to \mathbb{R}$ such that $f(0)$ and $f(1)$ have opposite signs, find a zero of f.

Here's a realizer of the above problem: if f has a rational zero, output such a zero. Otherwise, run a bisection algorithm to find smaller and smaller dyadic intervals such that f takes opposite signs on their endpoints.

Is this realizer computable? Not as presented, because one cannot compute whether f has a rational zero. Nevertheless, notice that given some f, this realizer always outputs some f-computable real!

Hence IVT is pointwise computable.

Proposition 1.20. IVT *is not computable.*

Proof. For each $a \in [0,1]$, define the piecewise linear function g_a whose graph goes from $(0,-1)$ to $(1/3, a)$ to $(2/3, a)$ to $(1,1)$. We show that there is no computable function f such that for each $a \in [0,1]$, $f(a)$ is a zero of g_a.

Suppose otherwise. Then given a name p for 0, f must produce a name for some x such that $g_0(x) = 0$, i.e., $x \in [1/3, 2/3]$. Suppose that $x > 1/3$. By continuity, f must produce reals $> 1/3$, given any names which are sufficiently close to p. But then f is wrong for any names for $a > 0$ which are sufficiently close to p, since for $a > 0$, the only zero of g_a lies below $1/3$. Contradiction. If we assume that $x < 2/3$, we get a similar contradiction. □

Observe that in the above proof, we used a function g_0 which is constantly zero on an interval. In fact,

Theorem 1.21 (see [56, Theorem 6.3.7])**.** *The restriction of* IVT *to functions which are not constantly zero on any open interval is computable.*

1.4. **Reductions between problems.** Before formally defining reductions, let us see some examples which we hope to capture.

Example 1.22. Say that a problem P is a special case of another problem Q, i.e., every P-instance is a Q-instance, and every Q-solution to every P-instance is also a P-solution to said P-instance. Surely, we want P to be reducible to Q.

Example 1.23. IVT and WKL are related in the following way: suppose we are given a continuous function $f : [0,1] \to \mathbb{R}$ such that $f(0)$ and $f(1)$ have opposite signs. We can find a zero of f using the following bisection algorithm: at the beginning of stage n, we will have chosen an interval $[k/2^n, (k+1)/2^n]$ such that $f(k/2^n)$ and $f((k+1)/2^n)$ have different signs. There are three possible outcomes:

(1) f has different signs at the endpoints of $[2k/2^{n+1}, (2k+1)/2^{n+1}]$;
(2) f has different signs at the endpoints of $[(2k+1)/2^{n+1}, (2k+2)/2^{n+1}]$;
(3) $f((2k+1)/2^{n+1}) = 0$, at which point the algorithm stops.

This bisection algorithm is not computable, because the sign function $\mathrm{sgn} : \mathbb{R} \to \{0, -, +\}$ is discontinuous and hence not computable. But we can guess at the sign of a number: the initial guess is that it is 0, and if it is nonzero it must eventually reveal its sign.

This allows us to simulate the above algorithm using WKL: compute the binary tree T consisting of all σ such that at stage $|\sigma|$, for all $n < |\sigma|$, we have not seen that $f\left(\sum_{i<n} \frac{\sigma(i)}{2^{-(i+1)}}\right)$ and $f\left(2^{-n} + \sum_{i<n} \frac{\sigma(i)}{2^{-(i+1)}}\right)$ are both positive or both negative. It is easy to see that T has a string of every length, and is hence infinite. If P is a path on T, we claim that the real number x with binary expansion P is a zero of f. If not, by continuity of f, there is some σ which is an initial segment of P such that $f\left(\sum_{i<|\sigma|} \frac{\sigma(i)}{2^{-(i+1)}}\right)$ and $f\left(2^{-|\sigma|} + \sum_{i<|\sigma|} \frac{\sigma(i)}{2^{-(i+1)}}\right)$ are both positive or both negative. But then σ cannot extend to an infinite path, contradiction.

Example 1.24. WKL and HB_1 are related in the following way. Fix a computable bijection f from $2^{\mathbb{N}}$ to the Cantor middle-thirds set. Given an infinite subtree of $2^{<\mathbb{N}}$, we can enumerate its leaves. For each leaf σ, we enumerate an open interval whose intersection with the Cantor set is exactly $\{f(X) : X \text{ extends } \sigma\}$. We also enumerate intervals whose union is the complement of the Cantor set. Then f is a computable bijection between the reals which are not covered by these intervals and paths on the given tree.

Conversely, given an enumeration of open sets which does not cover $[0,1]$, we can dovetail it into an enumeration of rational open intervals with the same union. Consider the tree of all $\sigma \in 2^{<\mathbb{N}}$ such that the first $|\sigma|$ many rational open intervals does not cover the interval $\left[\sum_{i<|\sigma|} \frac{\sigma(i)}{2^{-(i+1)}}, 2^{-|\sigma|} + \sum_{i<|\sigma|} \frac{\sigma(i)}{2^{-(i+1)}}\right]$. Then any path on this tree computes a sequence of intervals shrinking rapidly to a real which is not covered by the given open sets.

Now, let us define Weihrauch reducibility on problems.

Definition 1.25. Given problems P and Q, we say that P is *Weihrauch reducible* (*strongly Weihrauch reducible*, respectively) to Q, written $P \leq_W Q$ ($P \leq_{sW} Q$, respectively), if there is a *forward functional* Γ and a *backward functional* Δ such that

(1) given a name p for a P-instance, $\Gamma(p)$ is a name for some Q-instance;
(2) if p is a name for a P-instance X, then for every name q for a Q-solution to $\Gamma(p)$, $\Delta(p \oplus q)$ ($\Delta(q)$, respectively) is a name for a P-solution to X.

Intuitively, $P \leq_W Q$ means that one can uniformly computably transform a realizer for Q into a realizer for P.

We mention here that a good reference for the theory of Weihrauch reducibility is the survey paper by Brattka, Gherardi, Pauly [11]. Their survey also contains extensive bibliographic remarks. In particular, see [11, pg. 11] for historical remarks about Weihrauch reducibility.

Remark 1.26. A Weihrauch reduction from P to Q is obligated to "use" Q. In particular, if there is a P-instance which does not compute any Q-instance, then P cannot be Weihrauch reducible to Q.

It is easy to see that $\leq_W$ is reflexive and transitive, so we can define the associated notion of Weihrauch equivalence and Weihrauch degrees: for multivalued functions P and Q, we say that P and Q are *Weihrauch equivalent*, written $P \equiv_W Q$, if $P \leq_W Q$ and $Q \leq_W P$. For a multivalued function P, its *Weihrauch degree* $\mathbf{p}$ is its $\equiv_W$-class. Weihrauch reducibility lifts to Weihrauch degrees in the usual way; that is, we say that $\mathbf{p} \leq_W \mathbf{q}$ if and only if there is some $P \in \mathbf{p}$ and $Q \in \mathbf{q}$ such that $P \leq_W Q$, if and only if for all $P \in \mathbf{p}$ and $Q \in \mathbf{q}$, we have $P \leq_W Q$. We will abuse notation and use $P \leq_W \mathbf{q}$ to mean that there is some $Q \in \mathbf{q}$ such that $P \leq_W Q$, or equivalently, for all $Q \in \mathbf{q}$, we have $P \leq_W Q$. We give $P \equiv_W \mathbf{q}$ the analogous meaning.

We can define strong Weihrauch equivalence and strong Weihrauch degrees in the same way.

Note the uniformity in the definition of Weihrauch reducibility: Γ and Δ have to satisfy the above conditions for all names for P-instances. In fact, Weihrauch reducibility on multivalued functions was independently rediscovered by Dorais, Dzhafarov, Hirst, Mileti, Shafer [21], who named it uniform reducibility.

Let us look at some examples. From before, we have the following examples of strong Weihrauch reductions:

– if P is a special case of Q, then $P \leq_{sW} Q$ (both backward and forward functionals are the identity);
– $\mathsf{IVT} \leq_{sW} \mathsf{WKL}$;
– $\mathsf{WKL} \equiv_{sW} \mathsf{HB}_1$.

Proposition 1.27. *A problem is Weihrauch reducible to* id *if and only if it is computable.*

Example 1.28. id is Weihrauch reducible but not strongly Weihrauch reducible to the constant problem $X \mapsto \emptyset$.

For an nontrivial example of a problem which is Weihrauch reducible but not strongly Weihrauch reducible to another problem, we turn to Ramsey's theorem. For each $n \in \mathbb{N}$, $[\mathbb{N}]^n$ denotes the set of size n subsets of $\mathbb{N}$, often called the set of n-tuples.

Definition 1.29. Define the problem RT^n_k corresponding to Ramsey's theorem for n-tuples and k colors: given a coloring $c : [\mathbb{N}]^n \to k$, output an infinite c-homogeneous set H, i.e., $c \upharpoonright [H]^n$ is constant.

In particular, RT^2_2 can be thought of as the following problem: given an infinite undirected graph on $\mathbb{N}$, output either an infinite clique or anti-clique.

Example 1.30. $\mathsf{RT}^1_2 \not\leq_W$ id: given any pair of functionals Γ and Δ, we will construct an instance $c : \mathbb{N} \to 2$ of RT^1_2 which witnesses that Γ and Δ do not form a Weihrauch reduction from RT^1_2 to id. In other words, we will construct $c : \mathbb{N} \to 2$ such that $\Delta(c \oplus \Gamma(c))$ is not an infinite c-homogeneous set.

We define one value of c at each step. Keep defining $c(n) = 0$ until $\Delta(c \oplus \Gamma(c))$ converges at some number. (If $\Delta(c \oplus \Gamma(c))$ never converges at some number, then we can take c to be the constant coloring 0.) Then we switch to defining $c(n) = 1$ forever, ensuring that $\Delta(c \oplus \Gamma(c))$ cannot be an infinite c-homogeneous set. (By the use principle, any number on which $\Delta(c \oplus \Gamma(c))$ has converged must be colored 0.)

Proposition 1.31. $\mathsf{RT}^{n_0}_{k_0} \leq_{sW} \mathsf{RT}^{n_1}_{k_1}$ *if* $n_0 \leq n_1$ *and* $k_0 \leq k_1$.

Proof. Given $c : [\mathbb{N}]^{n_0} \to k_0$, define $d : [\mathbb{N}]^{n_1} \to k_1$ by

$$d(x_0, \ldots, x_{n_1-1}) = c(x_0, \ldots, x_{n_0-1}).$$

Then any infinite d-homogeneous set is also c-homogeneous. □

Example 1.32. id $\leq_W \mathsf{RT}^1_2$ but id $\not\leq_{sW} \mathsf{RT}^1_2$. To prove the latter, suppose that Γ and Δ witness that id $\leq_{sW} \mathsf{RT}^1_2$. Consider any two inputs for Γ, e.g., 0^∞ and 1^∞. Both $\Gamma(0^\infty)$ and $\Gamma(1^\infty)$ must be 2-colorings. Then, there is a common RT^1_2-solution B of $\Gamma(0^\infty)$ and $\Gamma(1^\infty)$. But $\Delta(B)$ cannot be equal to both 0^∞ and 1^∞, contradiction.

In fact, we can push this much further. Any noncomputable instance of id witnesses that id $\not\leq_{sW}$ RT^1_2 in a strong way:

Theorem 1.33 (Dzhafarov, Jockusch [27]). *If A is noncomputable and $c : \mathbb{N} \to 2$ is any coloring, then there is an RT^1_2-solution for c which does not compute A.*

Their proof uses techniques of Cholak, Jockusch, Slaman [17]. For now, we will prove a slightly weaker statement: if A is noncomputable, $c : \mathbb{N} \to 2$ is a coloring, and Δ is a functional, then there is an RT^1_2-solution B to c such that $\Delta(B) \neq A$.

Proof. Case 1. If c is unbalanced (i.e., only finitely many numbers are colored 0, or only finitely many numbers are colored 1), we are done, because c has a computable RT^1_2-solution.

Case 2. Consider the class P of all $S \subseteq \mathbb{N}$ such that:

- for every $x \in \mathbb{N}$ and every finite $F_0, F_1 \subseteq S$, it is **not** the case that $\Delta^{F_0}(x)\downarrow \neq \Delta^{F_1}(x)\downarrow$;
- for every $x \in \mathbb{N}$ and every finite $F_0, F_1 \subseteq \overline{S}$, it is **not** the case that $\Delta^{F_0}(x)\downarrow \neq \Delta^{F_1}(x)\downarrow$.

P is a Π^0_1 class.

Case 2a. P is nonempty. By the cone-avoiding basis theorem, we can choose some $S \in P$ such that $A \not\leq_T S$. Take B to be any common RT^1_2-solution of c and S (we think of S as a 2-coloring).

Then $\Delta(B) \neq A$, for if $\Delta(B) = A$, then we can compute A using S as follows: for each x, search for finite $F \subseteq S$ (or $\overline{S}$, depending on whether $B \subseteq S$ or $B \subseteq \overline{S}$) such that $\Delta^F(x)\downarrow$. Then we must have $\Delta^F(x)\downarrow = A$.

Case 2b. P is empty. In particular, let C be the set of numbers which are colored 0 by c. C does not lie in P. If there is some $x \in \mathbb{N}$ and some finite $F_0, F_1 \subseteq C$ such that $\Delta^{F_0}(x)\downarrow \neq \Delta^{F_1}(x)\downarrow$, we can pick $F = F_0$ or F_1 such that $\Delta^F(x)\downarrow \neq A(x)$. Then take B to be $F \cup (C \backslash [0, \max F])$. Otherwise, there is some $x \in \mathbb{N}$ and some finite $F_0, F_1 \subseteq \overline{C}$ such that $\Delta^{F_0}(x)\downarrow \neq \Delta^{F_1}(x)\downarrow$. We proceed similarly. □

Another class of problems of interest is stable Ramsey's theorem:

Definition 1.34. Let SRT^n_k denote the restriction of RT^n_k to *stable* colorings, i.e., colorings $c : [\mathbb{N}]^n \to k$ such that for all $A \in [\mathbb{N}]^{n-1}$, $\lim_n c(A \cup \{n\})$ exists.

Let COH be the following problem: given an array $(R_i)_i$, produce a cohesive set C, i.e., for all i, either $C \subseteq^* R_i$ or $C \subseteq^* \overline{R_i}$.

SRT^2_2 and COH play a crucial role in Cholak, Jockusch, Slaman's [17] computability-theoretic and proof-theoretic analysis of RT^2_2. Observe that RT^2_2 can be solved by applying COH and then SRT^2_2 (more on this later when we discuss compositions of problems).

Observe that $\mathsf{RT}^1_k \leq_{sW} \mathsf{SRT}^2_k$: given a coloring $c : \mathbb{N} \to k$, define $d(m,n) = c(m)$. Then d is stable and any d-homogeneous set is also c-homogeneous. On the other hand:

Proposition 1.35 (Hirschfeldt, Jockusch [36])**.** $\mathsf{RT}^1_{k+1} \not\leq_W \mathsf{SRT}^2_k$.

Proof. Given any pair of functionals Γ and Δ, we will construct an instance c of RT^1_{k+1} and an SRT^2_k-solution to Γ^c which witnesses that Γ and Δ do not form a Weihrauch reduction from RT^1_{k+1} to SRT^2_k.

We construct c in stages. At stage s, for each $j < k$, we search for the least finite set F_j (if any) such that:

- at this point in time, F_j appears to be an initial segment of a Γ^c-homogeneous set of color j (i.e., F_j is Γ^c-homogeneous of color j and for each $x \in F_j$, $\Gamma^c(x,y) = j$, where y is the largest number such that $\Gamma^c(x,y)$ is defined);
- $\Delta^{c \oplus F_j}(n)\downarrow = 1$ for some $n < s$.

If such F_j exists, define i_j to be $c(n_j)$, where n_j is least such that $\Delta^{c\oplus F_j}(n_j)\downarrow = 1$. Finally, we define $c(s)$ to be the least color which is not equal to any i_j.

Now consider the stable coloring $\Gamma^c : [\mathbb{N}]^2 \to k$. Let H be an infinite Γ^c-homogeneous set, say of color $j < k$. Then $\Delta^{c\oplus H}$ is an infinite c-homogeneous set. Since H is Γ^c-homogeneous of color j and $\Delta^{c\oplus H}$ is nonempty, it follows that the search for F_j during our construction of c must terminate. Hence the search stabilizes, with some eventual F_j. Let n_j be least such that $\Delta^{c\oplus F_j}(n_j)\downarrow = 1$.

But now F_j extends to an infinite Γ^c-homogeneous set H' (e.g., take the union of F_j and every element of H greater than $\max F_j$ which has color j with every element of F_j). $\Delta^{c\oplus H'}$ cannot be an infinite c-homogeneous set because it contains n_j, yet we never color c with color $c(n_j)$ after the search for F_j stabilizes. Contradiction. $\square$

Next, we define some nonuniform notions of reducibility.

Definition 1.36 (Dzhafarov [23])**.** Given problems P and Q, we say that P is *computably reducible* (*strongly computably reducible* resp.) to Q, written $P \leq_c Q$ ($P \leq_{sc} Q$ resp.), if every P-instance X computes some Q-instance Y such that for every Q-solution Z to Y, $X \oplus Z$ (Z resp.) computes a P-solution to X.

Example 1.37. Being pointwise computable is equivalent to being computably reducible to id. In particular, $\mathsf{IVT} \leq_c \mathrm{id}$.

Example 1.38. For each k, $\mathsf{RT}^1_k \leq_{sc}$ id: given a coloring $c : \mathbb{N} \to k$, take c to be an instance of id. Given c, one can nonuniformly compute an infinite c-homogeneous set (simply fix a color which appears infinitely often and take all elements with that color).

On the other hand, we showed earlier that $\mathsf{RT}^1_2 \not\leq_W$ id.

Example 1.39. Dzhafarov, Patey, Solomon, Westrick [28] showed that $\mathsf{RT}^1_3 \not\leq_{sc} \mathsf{SRT}^2_2$. This is an example of two problems which are $\leq_c$ but not $\leq_W$ or $\leq_{sc}$.

Yet another notion of reducibility is *generalized Weihrauch reducibility*, introduced by Hirschfeldt, Jockusch [36, §4.2].

This concludes the basic setup for the framework of Weihrauch reducibility (and other computable reducibilities).

1.5. **Reverse mathematics.** Another framework which can be used to classify the strength of theorems is reverse mathematics. We briefly present the framework of reverse mathematics, in order to compare and contrast it with the framework of computable reducibilities.

Reverse mathematics begins with the maxim "When the theorem is proved from the right axioms, the axioms can be proved from the theorem." (Friedman, ICM 1974 [29]) In this case, the axioms would be necessary for proving the theorem! This maxim is justified by the remarkable "Big Five" phenomenon: in the decades since, it was found that many basic theorems in algebra, analysis, combinatorics, topology, etc. are provably *equivalent* to one of five systems of axioms, over the base system of RCA_0 (defined below). Furthermore, these five systems are *linearly ordered* in terms of provability. The standard reference for reverse mathematics is Simpson [54]. An excellent introduction to reverse mathematics, with emphasis on computable combinatorics, is Hirschfeldt [35].

The basic setup is as follows. First, we fix a language which is sufficiently expressive for formalizing our theorems of interest. The language of set theory certainly suffices, but in fact the language L_2 of second-order arithmetic (defined below) is already rich enough to formalize many theorems of interest. This includes most theorems about countable objects, and objects that can be represented by countable objects, such as the real numbers. Most of reverse mathematics has been conducted in L_2. (A notable exception is higher order reverse mathematics, initiated by Kohlenbach [43].)

Definition 1.40. L_2 consists of the usual language of first-order arithmetic, augmented with set variables and quantifiers over them, and a binary predicate symbol $\in$, relating numbers and sets. We also have the equality symbol relating sets, which always satisfies extensionality. An L_2-*structure* is a tuple

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

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

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

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

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

Definition 1.41. Apart from basic axioms asserting that $(\mathbb{N}, +, \cdot, 0, 1, <)$ is a discretely ordered commutative semiring, RCA_0 consists of:

- the Σ^0_1 induction axiom schema:
$$\varphi(0) \wedge (\varphi(n) \to \varphi(n+1)) \to \forall n \varphi(n),$$
for any $\varphi(n)$ which is Σ^0_1;
- the Δ^0_1 (recursive) comprehension axiom schema:
$$\forall n(\varphi(n) \leftrightarrow \neg\psi(n)) \to \exists X \forall n(n \in X \leftrightarrow \varphi(n)),$$
for any $\varphi(n)$ and $\psi(n)$ which are Σ^0_1.

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

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

lies in M as well.

Note also that we work in classical logic. (The study of reverse mathematics over intuitionistic logic is known as constructive reverse mathematics, see Diener [20].) In particular, proofs in RCA_0 can have complicated case divisions. For example, the bisection proof of IVT that we described earlier, which involves a case division into whether the given function has a rational zero, can be formalized in RCA_0.

Having fixed a base theory, our next step is to fix a theorem P, and investigate what axioms we need to add to our base theory in order to prove P. There are two directions to this investigation. First we need to find a sufficiently strong system T (typically consisting of set existence axioms, such as comprehension axioms) such that T (plus our base theory) proves P. After doing so, ideally, we want to obtain a *reversal*, i.e., we want to show that P (plus our base theory) proves T. That shows that the axioms T are both sufficient and necessary in order to prove P.

We have already defined one system from the Big Five, namely RCA_0, which also serves as our base theory. We present the other four systems in order of increasing strength. The next step up is WKL_0, which consists of RCA_0 and Weak König's Lemma: every infinite subtree of $2^{<\mathbb{N}}$ has an infinite path. Some theorems which are equivalent to WKL_0 are:

- every open cover of $[0, 1]$ has a finite subcover (Heine-Borel);
- every continuous function from $[0, 1]$ to $\mathbb{R}$ attains a supremum;
- every countable commutative ring has a prime ideal.

Next we have ACA_0, which consists of RCA_0 together with the Arithmetical Comprehension Axiom scheme: $\exists X \forall n(n \in X \leftrightarrow \varphi(n))$, for any $\varphi(n)$ which is arithmetical. Some theorems which are equivalent to ACA_0 are:

- every infinite finitely branching tree has an infinite path (König's lemma);
- every bounded sequence in $\mathbb{R}$ has a cluster point (Bolzano-Weierstrass);
- every countable commutative ring has a maximal ideal;
- Ramsey's theorem RT^n_k for $n \geq 3$ and $k \geq 2$.

Yet another system in the Big Five is Arithmetical Transfinite Recursion (ATR_0), which consists of RCA_0 together with an axiom stating that one can iterate arithmetical comprehension along any well-ordering. It is important to note that being a well-ordering is not absolute for models of second-order arithmetic, hence in a particular model of ATR_0, one may be able to iterate arithmetical comprehension along ill-founded linear orderings which appear well-founded. Some theorems which are equivalent to ATR_0 are:

- any two countable well-orderings are comparable;

– any uncountable closed subset of $\mathbb{R}$ has a perfect subset;
– the open Ramsey theorem.

Finally, the strongest system in the Big Five is Π^1_1-Comprehension (Π^1_1-CA_0), which consists of RCA_0 together with the Π^1_1-Comprehension scheme: $\exists X \forall n(n \in X \leftrightarrow \varphi(n))$, for any $\varphi(n)$ which is Π^1_1. This system is equivalent to the Cantor-Bendixson theorem: every closed set in $\mathbb{R}$ is the union of a perfect closed set and a countable set.

There is a connection between proof-theoretic strength over RCA_0 and computability-theoretic strength. We say that a model of second-order arithmetic is an *ω-model* if its first-order universe is the standard natural numbers (usually denoted by ω). Then:

– The ω-models of RCA_0 are exactly the Turing ideals, i.e., the subsets of $\mathcal{P}(\omega)$ which are closed under $\oplus$ and $\leq_T$.
– The ω-models of ACA_0 are exactly the Turing ideals which are closed under the Turing jump.
– Every ω-model of ATR_0 is closed under hyperarithmetic reduction.

One often establishes relationships between a Π^1_2 statement and the Big Five using computability-theoretic methods. Fix a theorem P of the form $\forall A(\Theta(A) \to \exists B \Lambda(A, B))$, where Θ and Φ are arithmetic formulas[1]. If A satisfies $\Theta(A)$, then we say that it is a P-instance. If A is a P-instance and B satisfies $\Lambda(A, B)$, then we say that B is a P-solution to A. Then:

– If there is a computable P-instance with no computable P-solution, then P is not provable in RCA_0.
– If there is a computable P-instance with no low P-solution, then P is not provable in WKL_0 (using the Low Basis Theorem).
– If there is a computable P-instance with no arithmetical P-solution, then P is not provable in ACA_0.

Such methods can be used to establish reversals as well. For example, if one constructs a computable P-instance such that every P-solution computes $\emptyset'$, then this usually yields a proof that P implies ACA_0.

The endeavor of demonstrating nonimplications in reverse mathematics (specifically over ω-models) motivates the notion of computable reduction. Suppose we want to construct an ω-model $\mathcal{M}$ which satisfies P but not Q. (Note that by absoluteness of arithmetical statements for ω-models, being a P-instance or P-solution is absolute. Likewise for Q.) Then, there is some Q-instance X in $\mathcal{M}$ such that $\mathcal{M}$ contains

[1] If Θ and Φ are not arithmetic formulas then we have to be careful because being a P-instance or P-solution may not be absolute (even for ω-models)!

no Q-solution to X. Since any P-instance which is computable in X (and hence lies in $\mathcal{M}$) must have some P-solution in $\mathcal{M}$, this means that the Q-instance X witnesses that $Q \not\leq_c P$.

In fact, showing that $Q \not\leq_c P$ is a first step towards constructing an ω-model $\mathcal{M}$ which satisfies P but not Q. For example, a common method of constructing such $\mathcal{M}$ proceeds by constructing a Q-instance X with no X-computable Q-solution, such that for any set A_0 such that X has no A_0-computable Q-solution, and any A_0-computable P-instance Y, there is some P-solution A_1 of Y such that X has no $(A_0 \oplus A_1)$-computable Q-solution. (The special case $A_0 = X$ corresponds to $Q \not\leq_c P$.) Iterating this result shows that the ω-model consisting of all $(\bigoplus_i A_i)$-computable sets satisfies P but not Q.

Next, we discuss some differences between the frameworks of reverse mathematics and computable reducibilities.

(1) Resource sensitivity: Proofs in reverse mathematics can apply their premises multiple times in parallel or in series, but computable reductions or Weihrauch reductions can only use them once. Example: Ramsey's theorem for different number of colors (see below).
(2) Uniformity: Weihrauch reductions can be used to measure the uniform computational content of problems, while proofs in reverse mathematics can have nonuniform case divisions. Example: IVT is provable in RCA_0 but it is not a computable problem.
(3) Invariance under logical equivalence: Logically equivalent statements can correspond to problems which are not equivalent under computable reductions, depending on what we view as an instance and what we view as a solution. Example: HB_0 and HB_1 have different computational strength.
(4) "Burden of proof": Proofs in reverse mathematics are only allowed to use certain axioms, while computable reductions can be constructed using the full metatheory. To quote Gherardi, Marcone [32]:

   > [..] the computable analyst is allowed to conduct an unbounded search for an object that is guaranteed to exist by (nonconstructive) mathematical knowledge, whereas the reverse mathematician has the burden of an existence proof with limited means.

   Example: RCA_0 does not prove that our algorithm for computing HB_0 terminates. That requires WKL_0. This is reflected in the fact that $\mathsf{HB}_1 \equiv_W \mathsf{WKL}$.

This phenomenon also arises when we consider concepts that are not arithmetical (and hence not absolute for models of second-order arithmetic). For example, if a linear ordering L is ill-founded but all of its descending sequences are complicated, then a model could think that L is well-founded simply because none of its descending sequences lie in the model. We will discuss this more when we discuss ATR later in the course.

(5) Objects of study: Meaningful problems may not correspond to meaningful theorems. Example: problems of the form "given a nonempty set, produce a point in it". These are known as choice problems. (We will see concrete definitions in the future.)

Usually, classifications in the Weihrauch lattice refine classifications in reverse mathematics.

Example 1.42. We can prove RT^2_4 using two applications of RT^2_2: given $c : [\mathbb{N}]^2 \to 4$, define $d_0 : [\mathbb{N}]^2 \to 2$ by

$$d_0(m,n) = \begin{cases} 0 & \text{if } f(m,n) = 0 \text{ or } 1 \\ 1 & \text{if } f(m,n) = 2 \text{ or } 3 \end{cases}.$$

Apply RT^2_2 to obtain an infinite d_0-homogeneous set C_0. Without loss of generality, suppose that $d_0 \upharpoonright [C_0]^2$ has range $\{0\}$. Then, we may apply RT^2_2 to $d_0 \upharpoonright [C_0]^2$ to obtain an infinite c-homogeneous set. This shows that RT^2_2 implies RT^2_4 over RCA_0 (of course, the converse holds as well).

The above proof does not translate into a Weihrauch reduction from RT^2_4 to RT^2_2, and indeed $\mathsf{RT}^2_4 \not\leq_W \mathsf{RT}^2_2$ (Hirschfeldt, Jockusch [36], Brattka, Rakotoniaina [15]). In fact, Patey [46] showed that $\mathsf{SRT}^n_k \not\leq_c \mathsf{RT}^n_l$ for $n \geq 2$ and $k > l \geq 2$.

Nevertheless, Hirst and Mummert [37] showed that we can prove RT^2_4 using one application of RT^2_2. Given $c : [\mathbb{N}]^2 \to 4$, consider the following two cases.

<u>Case 1.</u> There is an infinite set X such that c takes at most two colors on $[X]^2$. If so, fix such X and fix $\{a_0, a_1\} \subseteq 4$ which contains the range of $c \upharpoonright [X]^2$. Then define $d : [X]^2 \to 2$ by

$$d(m,n) = \begin{cases} 0 & c(m,n) = a_0 \\ 1 & c(m,n) = a_1 \end{cases}.$$

Apply RT^2_2 to d to obtain an infinite c-homogeneous set.

Case 2. If not, define $d : [X]^2 \to 2$ by

$$d(m,n) = \begin{cases} 0 & c(m,n) = 0 \text{ or } 1 \\ 1 & c(m,n) = 2 \text{ or } 3 \end{cases}.$$

Apply RT^2_2 to d to obtain an infinite d-homogeneous set X. But then c takes at most 2 colors on $[X]^2$, contradiction. Hence Case 2 cannot occur.

The above proof appears to use RT^2_2 twice, but with a little work we can define a single coloring d which does the job, instead of defining d in both Case 1 and 2.

Hirst and Mummert's proof shows us a difficulty in defining a meaningful notion of "the number of times you use a theorem in a proof". The above proof relies heavily on the use of the law of excluded middle. In fact, working in the intuitionistic higher-order system $i\mathsf{RCA}^\omega_0$, Hirst and Mummert [37] establish an equivalence between proofs which "use their premise once" and formal Weihrauch reducibility.

2. Operations on problems and their algebraic properties

There are several natural operations on problems, which lift to corresponding operations on the Weihrauch degrees. Here are some reasons to study them:

- to investigate whether the Weihrauch lattice models some logic (see Brattka, Gherardi [9]);
- to provide precise calibrations of problems of interest (e.g., a problem may not be equivalent to any known problem, but it could be equivalent, or at least reducible, to a product of two known problems);
- to aid in establishing reductions and nonreductions between problems (we will see examples in due course).

Here are some basic operations:

- Composition: $f \circ g$ has instances $\{X : \forall Y \in g(X)[f(Y) \neq \emptyset]\}$, and each instance X has solution set $f(g(X))$.
- Coproduct: $f_0 \sqcup f_1$ has instances $\bigcup_{i=0,1}\{(i,X) : X \text{ is an } f_i\text{-instance}\}$. For $i = 0,1$, (i,Y) is a $(f_0 \sqcup f_1)$-solution to (i,X) if Y is a f_i-solution to X.
- Meet: $f_0 \sqcap f_1$ has instances $\{(X_0,X_1) : X_i \text{ is a } f_i\text{-instance}\}$. For $i = 0,1$, (i,Y) is a $(f_0 \sqcap f_1)$-solution to (X_0,X_1) if Y is a f_i-solution to X_i.

– Parallel product: $f_0 \times f_1$ has instances $\{(X_0, X_1) : X_i \text{ is a } f_i\text{-instance}\}$. (Y_0, Y_1) is a $(f_0 \times f_1)$-solution to (X_0, X_1) if for each $i = 0, 1$, Y_i is an f_i-solution to X_i.
– Finite parallelization: f^* has instances which are sequences of f-instances of any finite length, with f^*-solutions being a sequence of f-solutions for each given f-instance.
– (Infinite) parallelization: $\widehat{f}$ has instances which are $\mathbb{N}$-sequences of f-instances, with $\widehat{f}$-solutions being a sequence of f-solutions for each given f-instance.

It is easy to see that all of the above operations, except composition, are monotone with respect to both $\leq_W$ and $\leq_{sW}$. Hence they lift to the Weihrauch degrees and the strong Weihrauch degrees. Now we address some natural questions about the Weihrauch degrees:

– There is a least Weihrauch degree, consisting of the problems with empty domain.
– There is *no* greatest Weihrauch degree, if we assume choice (which implies that every problem has a realizer). See Brattka, Pauly [13, §2.1] for details. In these notes, we always assume the axiom of choice.
– id is the identity with respect to the coproduct, meet, and parallel product on the Weihrauch degrees.
– The problems above id are known as *pointed*. They are exactly the problems with some computable instance. (Recall that in $\leq_W$ or $\leq_{sW}$, you are obliged to make use of the given problem.)
– The Weihrauch degrees with $\sqcap$ and $\sqcup$ form a distributive lattice (Pauly [48]), i.e., meets and joins are distributive.
– The coproduct operation is *not* a join in the strong Weihrauch degrees. Nevertheless, the strong Weihrauch degrees form a (nondistributive) lattice (Dzhafarov [25]).

2.1. **Parallel product.** First, a basic fact:

Proposition 2.1. *If every f-instance computes some g-instance, then $f \leq_W f \times g$. It follows that if f and g are pointed, then $f \sqcup g \leq_W f \times g$.*

The parallel product characterizes when $\leq_W$ and $\leq_{sW}$ agree:

Proposition 2.2. *For any problem f, $\mathrm{id} \times f \leq_{sW} f$ if and only if for all problems g,*

$$g \leq_W f \quad \Leftrightarrow \quad g \leq_{sW} f.$$

Proof. ($\Rightarrow$). Suppose that $\mathrm{id} \times f \leq_{sW} f$ and $g \leq_W f$. Fix functionals Γ_0 and Δ_0 witnessing that $\mathrm{id} \times f \leq_{sW} f$. Fix functionals Γ_1 and Δ_1 witnessing that $g \leq_W f$.

We show that $g \leq_{sW} f$: given a g-instance X, compute the $(\mathrm{id} \times f)$-instance $(X, \Gamma_1(X))$. Then compute the f-instance $\Gamma_0(X, \Gamma_1(X))$. If Y is an f-solution to $\Gamma_0(X, \Gamma_1(X))$, then $(\Delta_0(Y))_0 = X$ and $(\Delta_0(Y))_1$ is an f-solution to $\Gamma_1(X)$, so $\Delta_1(\Delta_0(Y))$ is a g-solution to X.

($\Leftarrow$). Take $g = \mathrm{id} \times f$. □

Definition 2.3. A problem f is called a *cylinder* if $\mathrm{id} \times f \leq_{sW} f$.

Proposition 2.4. *Every problem f is Weihrauch equivalent to a cylinder, e.g.,* $\mathrm{id} \times f$.

Definition 2.5. Define the *limit* problem for a computable metric space X, written $\lim_X$, as follows: given a convergent sequence in X, output its limit.

We denote $\lim_{\mathbb{N}^\mathbb{N}}$ by lim.

Note that $\lim_X$ is single-valued.

Example 2.6. lim and WKL are cylinders. RT^1_2 is not a cylinder (we showed earlier that $\mathrm{id} \not\leq_{sW} \mathsf{RT}^1_2$).

It is often useful to know when one is working with cylinders, because a Weihrauch reduction to a cylinder yields a strong Weihrauch reduction, and conversely, a strong Weihrauch nonreduction to a cylinder yields a Weihrauch nonreduction.

2.2. **Finite and infinite parallelization.** Let us begin with the infinite parallelization, since it occurs much more often. It is not hard to show that parallelization is a closure operator with respect to $\leq_W$, i.e.,

$$f \leq_W \widehat{f}, \qquad \widehat{\widehat{f}} \leq_W \widehat{f}, \qquad \text{and} \qquad (f \leq_W g \to \widehat{f} \leq_W \widehat{g}).$$

It is also a closure operator with respect to $\leq_{sW}$.

Example 2.7. lim and WKL are closed under parallelization.

Note that closure under parallelization implies closure under parallel product. We will see later that the converse fails (choice on natural numbers $\mathsf{C}_\mathbb{N}$). Speaking of choice:

Definition 2.8. Define the problem of *closed choice* in a computable metric space X, written C_X, as follows: given a closed set (represented negatively), choose an element of the closed set.

Closed choice is the most common type of choice problem that comes up, so we will simply refer to it as choice.

Example 2.9. $\mathsf{WKL} \equiv_{sW} \mathsf{C}_{2^\mathbb{N}}$ and $\mathsf{HB}_1 \equiv_{sW} \mathsf{C}_{[0,1]}$.

Proposition 2.10 (essentially Brattka, Presser [14]). C_X *is strongly Weihrauch equivalent to the problem of finding a zero of a continuous function* $f : X \to \mathbb{R}$.

Proof. ($\leq_{sW}$.) Given an enumeration of rational open balls $(B(q_i, r_i))_i \subseteq X$, we want to compute a continuous function $f : X \to \mathbb{R}$ such that $f^{-1}(0) = X \setminus \bigcup_i B(q_i, r_i)$. Define

$$f(x) = \sum_{i=0}^{\infty} \frac{\max\{0, r_i - d(q_i, x)\}}{r_i} \cdot 2^{-i-1}.$$

Observe that $f^{-1}(0) = X \setminus \bigcup_i B(q_i, r_i)$ and that f is computable in $(B(q_i, r_i))_i$.

($\geq_{sW}$.) Given a continuous function $f : X \to \mathbb{R}$ with a zero, we want to enumerate rational open balls $(B_i)_i \subseteq X$ such that $f^{-1}(0) = X \setminus \bigcup_i B_i$. We can do so because for each q_i, if $f(q_i) \neq 0$, then we will eventually discover that, together with an open ball about q_i whose image under f does not contain 0. (By feeding f arbitrarily long initial segments of a name of q_i, we must obtain arbitrarily long initial segments of a name of $f(q_i)$.) □

There are several useful variants of closed choice; we will define them as required.

Next, we give our first example of a problem which is equivalent to the parallelization of another problem.

Proposition 2.11 (Brattka, Gherardi [8]). $\widehat{\mathsf{C}_2} \equiv_{sW} \widehat{\mathsf{IVT}} \equiv_{sW} \mathsf{WKL}$.

Proof. First, we show that $\mathsf{C}_2 \leq_{sW} \mathsf{IVT}$. For each $a \in [0, 1]$, consider the piecewise linear function g_a whose graph goes from $(0, -1)$ to $(1/3, a)$ to $(2/3, a)$ to $(1, 1)$. Given a C_2-instance, if it enumerates $i \in \{0, 1\}$ at stage n, then we compute $g_{(-1)^i 2^{-n}}$. Otherwise we compute g_0.

Given a zero x of g_a (as computed above), x will eventually reveal that $x > 1/3$ or $x < 2/3$. If $x > 1/3$, then a cannot be 2^{-n}, so 0 is a solution to the given C_2-instance. Similarly, if $x < 2/3$, then 1 is a solution to the given C_2-instance.

It follows that $\widehat{\mathsf{C}_2} \leq_{sW} \widehat{\mathsf{IVT}}$.

Next, we showed earlier that $\mathsf{IVT} \leq_{sW} \mathsf{WKL}$. So $\widehat{\mathsf{IVT}} \leq_{sW} \widehat{\mathsf{WKL}} \leq_{sW} \mathsf{WKL}$.

Finally, we show that $\mathsf{WKL} \leq_{sW} \widehat{\mathsf{C}_2}$: given some infinite $T \subseteq 2^{<\mathbb{N}}$, one might think of using C_2 to choose, for each $\sigma \in T$, some i such that $\sigma^\frown i$ is extendible. But for nonextendible σ, neither $\sigma^\frown 0$ or $\sigma^\frown 1$ are extendible, so this is not an instance of C_2.

Instead, for each $\sigma \in T$, consider

$$\{i \in 2 : \forall n(\sigma^\frown(1-i) \text{ has an extension of length } n \\ \to \sigma^\frown i \text{ has an extension of length } n)\}.$$

For each $\sigma \in T$, the above set is Π^0_1 (in T) and nonempty. Hence it can be thought of as an instance of C_2. Given a solution for each such instance of C_2, we can compute an infinite path on T. □

Definition 2.12. LPO is the following (single-valued) function: given $p \in \mathbb{N}^\mathbb{N}$, $\mathsf{LPO}(p) = 0$ if $p(n) = 0$ for all n, otherwise $\mathsf{LPO}(p) = 1$.

LPO corresponds to the limited principle of omniscience, which is a weak form of the law of excluded middle, studied in constructive mathematics.

Note that $\mathsf{C}_2 \leq_W \mathsf{LPO}$. This reduction is strict, as we will see below.

Proposition 2.13. $\widehat{\mathsf{LPO}}$ *is a cylinder.*

Proof. It suffices to observe that $\mathrm{id} \leq_{sW} \widehat{\mathsf{LPO}}$, because we then have that $\mathrm{id} \times \widehat{\mathsf{LPO}} \leq_{sW} \widehat{\mathsf{LPO}} \times \widehat{\mathsf{LPO}} \leq_{sW} \widehat{\mathsf{LPO}}$. □

Proposition 2.14 (Brattka, Gherardi [7])**.** $\lim \equiv_{sW} \widehat{\mathsf{LPO}} \equiv_{sW} \widehat{\mathsf{C}_\mathbb{N}}$.

Proof. $\lim \leq_{sW} \widehat{\mathsf{LPO}}$: Since $\widehat{\mathsf{LPO}}$ is a cylinder, it suffices to show that $\lim \leq_W \widehat{\mathsf{LPO}}$. Given $(p_n)_n$, for each m and k, define an LPO-instance as follows:

$$q_{m,k}(n) = \begin{cases} 0 & \text{if } p_m(k) = p_{m+n}(k) \\ 1 & \text{otherwise} \end{cases}.$$

Given answers to all of these LPO-instances, we can compute $\lim_n p_n(k)$ as follows: search for sufficiently large m such that $\mathsf{LPO}(q_{m,k}) = 0$. Then $\lim_n p_n(k) = p_m(k)$.

$\widehat{\mathsf{LPO}} \leq_{sW} \widehat{\mathsf{C}_\mathbb{N}}$: It suffices to show that $\mathsf{LPO} \leq_{sW} \mathsf{C}_\mathbb{N}$. Given $p \in \mathbb{N}^\mathbb{N}$, we enumerate a proper subset of $\mathbb{N}$ as follows. At stage s, as long as $p(n) \neq 0$ for all $n \leq s$, we enumerate $s+1$. Otherwise, if $p(s) = 0$, we enumerate 0. Then:

- if $p(n) \neq 0$ for all n, then 0 is the only $\mathsf{C}_\mathbb{N}$-solution;
- otherwise, 0 is not a $\mathsf{C}_\mathbb{N}$-solution.

$\widehat{\mathsf{C}_\mathbb{N}} \leq_{sW} \lim$: It is easy to see that $\mathsf{C}_\mathbb{N} \leq_{sW} \lim$ (we show below that $\mathsf{C}_\mathbb{N} \equiv_{sW} \lim_\mathbb{N}$), so $\widehat{\mathsf{C}_\mathbb{N}} \leq_{sW} \widehat{\lim} \leq_{sW} \lim$ as desired. □

It follows that $\widehat{\mathsf{C}_2} \equiv_{sW} \mathsf{WKL} <_{sW} \lim \equiv_{sW} \widehat{\mathsf{LPO}}$, so $\mathsf{C}_2 <_W \mathsf{LPO}$. (See Brattka, Gherardi [8, Theorem 7.13] for a different proof.)

Since $\mathsf{C}_{\mathbb{N}}$ is pointwise computable, the above also shows that $\widehat{\mathsf{C}_{\mathbb{N}}}$ is not even computably reducible to $\mathsf{C}_{\mathbb{N}}$. Hence $\mathsf{C}_{\mathbb{N}}$ is not closed under parallelization. It is, however, closed under parallel product (exercise).

Proposition 2.15. $\mathsf{C}_{\mathbb{N}} \equiv_{sW} \max_{\mathbb{N}} \equiv_{sW} \lim_{\mathbb{N}}$.

Proof. $\mathsf{C}_{\mathbb{N}} \leq_{sW} \max_{\mathbb{N}}$: Given an enumeration of a proper subset S of $\mathbb{N}$, define a sequence $(m_i)_i$ as follows: m_i is the largest number which has not been enumerated by stage i. Then $\max_i m_i$ lies outside S.

$\max_{\mathbb{N}} \leq_{sW} \lim_{\mathbb{N}}$: Given a sequence $(m_i)_i$, apply $\lim_{\mathbb{N}}$ to the sequence $(\max_{j \leq i} m_j)_i$.

$\lim_{\mathbb{N}} \leq_{sW} \mathsf{C}_{\mathbb{N}}$: Given a convergent sequence $(m_i)_i$, enumerate the complement of $\{\langle m, s\rangle : (\forall i > s)[m_i = m]\}$. If $\langle m, s\rangle$ is a $\mathsf{C}_{\mathbb{N}}$-solution, then $\lim_i m_i = m$. □

As for $\min_{\mathbb{N}}$, we can characterize it using finite parallelization.

Proposition 2.16. $\min_{\mathbb{N}} \equiv_{sW} \mathsf{LPO}^*$.

Proof. $\min_{\mathbb{N}} \leq_{sW} \mathsf{LPO}^*$: given $p \in \mathbb{N}^{\mathbb{N}}$, we construct $p(0)$-many instances of LPO as follows. For each $i < p(0)$, the i^{th} instance encodes whether i appears in p.

$\mathsf{LPO}^* \leq_{sW} \min_{\mathbb{N}}$: given instances $p_0, \ldots, p_{k-1}$ of LPO, we define $q \in \mathbb{N}^{\mathbb{N}}$ by $q(n) = \sum_{i<k} p_i(n) \cdot 2^i$. Then we can compute the answer to each LPO-instance using $\min_n q(n)$. □

Note that LPO^* is pointwise computable, hence lim (and equivalently, $\widehat{\mathsf{LPO}}$) is not even computably reducible to LPO^*.

Here are some nontrivial examples where finite parallelization has proven useful:

- characterizing the problem of finding Nash equilibria (Pauly [47]);
- characterizing some combinatorial principles which are equivalent (in reverse mathematics) to weak induction principles (Davis, Hirschfeldt, Hirst, Pardo, Pauly, Yokoyama [19]).

2.3. **Composition and compositional product.** Recall the definition of the composition of problems f and g: the problem $f \circ g$ has instances $\{X : \forall Y \in g(X)[f(Y) \neq \emptyset]\}$, and each instance X has solution set $f(g(X))$. Unlike the other operations we have introduced, the composition does not lift to an operation on Weihrauch degrees. In fact, note that $f \circ g$ only makes sense if the codomain of g and the domain of f are the same represented space. Whenever we write $f \circ g$ we will implicitly assume that this is the case.

Furthermore, the composition does not fully reflect how we compose problems in practice. In particular, we might want to modify the output of g before feeding it to f. This modification might use both the given input of g and the output of g.

Therefore, we need a new definition.

Definition 2.17. The *compositional product* of f and g, denoted $f * g$, is the maximum Weihrauch degree among $\{f_0 \circ g_0 : f_0 \leq_W f, g_0 \leq_W g\}$.

It is clear that whenever $f * g$ is defined, then it induces an operation on Weihrauch degrees, which is monotone in each component. It turns out that $f * g$ is always defined (Theorem 2.19). Before presenting that result, we mention the following basic property:

Proposition 2.18. *For any problems f and g, $f \times g \leq_W f * g$ whenever $f * g$ is defined (which, by Theorem 2.19, is always the case.)*

Proof. Observe that $f \times g \equiv_W (f \times \mathrm{id}) \circ (\mathrm{id} \times g)$.[2] Since $f \times \mathrm{id} \leq_W f$ and $\mathrm{id} \times g \leq_W g$, we have that $(f \times \mathrm{id}) \circ (\mathrm{id} \times g) \leq_W f * g$ as desired. $\square$

In other words, if one has the power to solve problems in series, then one can also solve them in parallel.

Theorem 2.19 (Brattka, Pauly [13])**.** *For any problems f and g, $f * g$ is always defined.*

Partial sketch. Suppose $f :\subseteq U \rightrightarrows V$ is a cylinder and $g :\subseteq X \rightrightarrows Y$ is a problem. (The proof for general f follows by replacing f with $\mathrm{id} \times f$.)

Consider $f_0 \circ g_0$, where $f_0 \leq_W f$ and $g_0 \leq_W g$. Since f is a cylinder, $f_0 \leq_{sW} f$. Fix Γ and Δ witnessing that $f_0 \leq_{sW} f$. Fix Φ and Ψ witnessing that $g_0 \leq_W g$. We claim that there is a computable *multivalued* function $\Theta :\subseteq \mathbb{N}^{\mathbb{N}} \times Y \rightrightarrows U$ such that

$$f_0 \circ g_0 \leq_{sW} f \circ \Theta \circ (\mathrm{id} \times g).$$

For the forward functional, we use $\mathrm{id} \times \Phi$. We define

$$\Theta = \delta_U \circ \Gamma \circ \Psi \circ (\mathrm{id} \times \delta_Y^{-1}).$$

Θ is computable because $\Gamma \circ \Psi$ realizes Θ. For the backward functional, we use Δ. This proves our claim.

Next, we want to define a problem $g^t \leq_W g$ such that for any computable multivalued function $\Theta :\subseteq \mathbb{N}^{\mathbb{N}} \times Y \rightrightarrows U$,

$$f \circ \Theta \circ (\mathrm{id} \times g) \leq_W f \circ g^t.$$

[2]In order for the types to match, we should say $(f^r \times \mathrm{id}) \circ (\mathrm{id} \times g^r)$, where if $f :\subseteq X \rightrightarrows Y$, then $f^r :\subseteq \mathbb{N}^{\mathbb{N}} \rightrightarrows \mathbb{N}^{\mathbb{N}}$ is defined to be $\delta_Y^{-1} \circ f \circ \delta_X$. f^r is called the *realizer version* of f; indeed f^r and f have the same realizers.

If g^t could take computable multivalued functions (such as Θ) as part of its input, then we could define $g^t(\Theta, p, x) = \Theta(p, g(x))$ and prove the above reduction. However, we do not have a "nice" representation for the space of computable (or continuous) multivalued functions. Instead, Brattka and Pauly [13] introduce the represented space of strongly computable (and strongly continuous) multivalued functions, and use it to complete the proof. $\square$

The (complete) proof of Theorem 2.19 yields a *cylindrical decomposition lemma*:

Lemma 2.20 (Brattka, Pauly [13]). *Let f and g be problems. If $F \equiv_W f$ and $G \equiv_W g$ are cylinders, then there is some computable function K such that $f * g \equiv_W F \circ K \circ G$.*

We proceed to consider some problems for which it is natural to consider their compositional product. Our first example comes from Cholak, Jockusch, Slaman's [17] splitting of RT^2_2 into its stable part SRT^2_2 and the cohesive principle COH.

Proposition 2.21. $\mathsf{RT}^2_2 \leq_W \mathsf{SRT}^2_2 * \mathsf{COH}$.

Proof. Given $c : [\mathbb{N}]^2 \to 2$, apply COH to the array $(R_a)_a$ where R_a is defined to be $\{b > a : c(a, b) = 0\}$. We obtain a cohesive set S. Then $c \upharpoonright [S]^2$ is stable: for each $a \in S$, $\lim_{b \in C, b \to \infty} c(a, b)$ is equal to 0 if $S \subseteq^* R_a$, and is equal to 1 if $S \subseteq^* \overline{R_a}$. Therefore we may apply SRT^2_2 to $c \upharpoonright [S]^2$ to obtain an infinite c-homogeneous set H.

Concretely, define P to be the problem whose instances are colorings $c : [\omega]^2 \to 2$, with solutions being stable colorings $c \upharpoonright S : [S]^2 \to 2$. Then P is Weihrauch reducible to COH, and the previous paragraph shows that $\mathsf{RT}^2_2 \leq_W \mathsf{SRT}^2_2 \circ \mathsf{P}$. Hence $\mathsf{RT}^2_2 \leq_W \mathsf{SRT}^2_2 * \mathsf{COH}$. $\square$

At Dagstuhl 2015, Brattka asked about the relationship between RT^2_2, $\mathsf{SRT}^2_2 * \mathsf{COH}$, and $\mathsf{SRT}^2_2 \times \mathsf{COH}$ up to Weihrauch reducibility.

Let NON denote the problem of producing a set which is not computable in the given set. Then:

Theorem 2.22 (Dzhafarov, Goh, Hirschfeldt, Patey, Pauly [26]). $\mathsf{LPO} \times \mathsf{NON} \not\leq_W \mathsf{RT}^2_2$.

In fact, the result holds for the restriction of NON to the instance $\emptyset$. Hence if Φ and Ψ form a Weihrauch reduction from LPO to RT^2_2, then there must be some instance S of LPO such that Φ^S is a coloring with some infinite computable homogeneous set.

A key notion in the proof of the above theorem is the following:

Definition 2.23. A coloring $c : [\omega]^2 \to 2$ is *balanced on* $X \subseteq \omega$ if for each $i < 2$, there is some infinite c-homogeneous subset of X of color i.

Theorem 2.24 (Jockusch [41])**.** *Let c be a computable 2-coloring. If an infinite set X does not contain any infinite c-homogeneous set of color i, then it must contain an infinite X-computable c-homogeneous set (of color $1-i$).*

Proof. We define a binary branching $\Pi^{0,X}_1$ class $\mathcal{C}$ in $\mathbb{N}^\mathbb{N}$ as follows. $f \in \mathbb{N}^\mathbb{N}$ lies in $\mathcal{C}$ if $f(0) = \min X$ and for each k, there is $i < 2$ such that $f(k+1)$ is the least number in X greater than $f(k)$ satisfying

- for all $j < k$, $c(f(j), f(j+1)) = c(f(j), f(k+1))$;
- $c(f(k), f(k+1)) = i$.

(Note that if $c(f(k), f(k+1)) = i$, then $c(f(k), f(l)) = i$ for all $l > k$.)

It is not hard to see that $\emptyset'''$ computes a path in $\mathcal{C}$, so $\mathcal{C}$ is nonempty[3]. Fix any f in $\mathcal{C}$. For $i < 2$, define $H_i = \{f(k) : c(f(k), f(k+1)) = i\}$. Then each H_i is homogeneous for color i.

Since we assumed that X does not contain any infinite c-homogeneous set of color 1, H_1 must be finite. Hence there is m such that for all $k > m$, $c(f(k), f(k+1)) = 0$. This shows that H_0 is X-computable. □

Lemma 2.25 ([26])**.** *Let $c : [\omega]^2 \to 2$ be computable, with no infinite computable c-homogeneous set. Then for any nonempty Π^0_1 class $\mathcal{C}$ consisting of 2-partitions of ω, there is some $\langle P_0, P_1\rangle$ in $\mathcal{C}$ and some $i < 2$ such that c is balanced on P_i.*

The proof of the above lemma uses a variant of Mathias forcing, with conditions being tuples $(E_{0,0}, E_{1,0}, E_{0,1}, E_{1,1}, X, \mathcal{D})$ satisfying:

- X is an infinite computable set;
- for each i, j, $\max E_{i,j} < \min X$;
- for every $x \in X$, c avoids the color i on $E_{i,j} \cup \{x\}$;
- $\mathcal{D} \subseteq \mathcal{C}$ is a nonempty Π^0_1 class such that for every $\langle P_0, P_1\rangle$ in $\mathcal{D}$, $E_{i,j} \subseteq P_j$.

Proof that $\mathsf{LPO} \times \mathsf{NON} \not\leq_W \mathsf{RT}^2_2$*.* Towards a contradiction, fix Φ and Ψ witnessing that $\mathsf{LPO} \times \mathsf{NON} \leq_W \mathsf{RT}^2_2$. We will build an instance S of LPO (S will be 0^ω or 0^n1^ω for some n) such that Φ and Ψ fail for the $(\mathsf{LPO} \times \mathsf{NON})$-instance $\langle S, \emptyset\rangle$.

For any LPO-instance S, $\Phi^{S\oplus\emptyset}$ is an RT^2_2-instance. For any infinite homogeneous set H for $\Phi^{S\oplus\emptyset}$, $\Psi^{S\oplus\emptyset\oplus H} = \{\mathsf{LPO}(S)\} \oplus Y$, where Y is some noncomputable set. We will show that we can always find some infinite $\Phi^{S\oplus\emptyset}$-homogeneous set H such that one of the following hold:

[3]Once we know that $\mathcal{C}$ is nonempty, we can obtain better complexity bounds. Jockusch [41] showed that $\mathcal{C}$ has a path which is low over $\emptyset'$.

(1) H is computable, in which case Y cannot be noncomputable;
(2) $S = 0^\infty$ and $\Psi^{0^\infty \oplus \emptyset \oplus H}(0) \simeq 0$ (i.e., it converges and equals 0 or it diverges);
(3) $S = 0^n 1^\infty$ for some n, and $\Psi^{0^n 1^\infty \oplus \emptyset \oplus H}(0)\downarrow = 1$.

To show that, let c denote the coloring $\Phi^{0^\infty \oplus \emptyset}$. Consider the Π^0_1 class $\mathcal{C}$ consisting of all 2-partitions $\langle P_0, P_1 \rangle$ of ω such that

$$(\forall i < 2)(\forall \text{ finite } F \subseteq P_i)[F \text{ is } c\text{-homogeneous of color } i \rightarrow \Psi^{0^\infty \oplus \emptyset \oplus F}(0) \simeq 0].$$

Case 1. Suppose $\mathcal{C}$ is nonempty. If there is some infinite c-computable homogeneous set, we satisfy (1), with $S = 0^\infty$. Otherwise, by Lemma 2.25, there is some $\langle P_0, P_1 \rangle$ in $\mathcal{C}$ and $i < 2$ such that c is balanced on P_i. Let $H \subseteq P_i$ be an infinite c-homogeneous set of color i. Then we satisfy (2), with $S = 0^\infty$.

Case 2. If $\mathcal{C}$ is empty, then by compactness, there is some m such that for every $\langle P_0, P_1 \rangle$, there is some $i < 2$ and some $F \subseteq P_i \upharpoonright m$ such that F is c-homogeneous of color i and $\Psi^{0^\infty \oplus \emptyset \oplus F}(0)\downarrow = 1$. Hence there is some u sufficiently large such that the above holds, with $\Psi^{0^u \oplus \emptyset \oplus F}(0)\downarrow = 1$. Then, we can take some $n > u$ such that $\Phi^{0^n \oplus \emptyset}$ agrees with c below u (hence the F's are $\Phi^{0^n \oplus \emptyset}$-homogeneous with the same color as before). Now, let $S = 0^n 1^\infty$, and let d denote the coloring $\Phi^{0^n 1^\infty \oplus \emptyset}$.

If d has an infinite computable homogeneous set, then we satisfy (1). Otherwise, we will show that we satisfy (3), by showing that one of the above F's extends to an infinite d-homogeneous set.

First, we may compute an infinite set B such that $\min B > m$ and for every $a < m$, $\lim_{b \in B} d(a, b)$ exists. (Construct B in m stages; at stage a, shrink B to whichever of $\{b : d(a, b) = 0\}$ or $\{b : d(a, b) = 1\}$ is infinite.) This lets us define a 2-partition of m: define $P_0 = \{a < m : \lim_{b \in B} d(a, b) = 0\}$, and define P_1 to be its complement.

Then there is some $i < 2$ and some $F \subseteq P_i \upharpoonright m$ such that F is d-homogeneous of color i and $\Psi^{0^n 1^\infty \oplus \emptyset \oplus F}(0)\downarrow = 1$. Since there is no infinite (B-)computable d-homogeneous set, B must contain an infinite d-homogeneous set H of color i. Then $F \cup H$ is an infinite d-homogeneous set such that $\Psi^{0^n 1^\infty \oplus \emptyset \oplus (F \cup H)}(0)\downarrow = 1$, i.e., we satisfy (3). $\square$

Since $\mathsf{LPO} \leq_W \mathsf{SRT}^2_2$ (trivially) and $\mathsf{NON} \leq_W \mathsf{COH}$ (Jockusch, Stephan [40] showed that every set which is cohesive for all primitive recursive sets is hyperimmune, and hence noncomputable), it follows that $\mathsf{SRT}^2_2 \times \mathsf{COH} \not\leq_W \mathsf{RT}^2_2$. In particular, $\mathsf{SRT}^2_2 * \mathsf{COH} \not\leq_W \mathsf{RT}^2_2$.

Note that RT^2_2 is not even computably reducible to $\mathsf{SRT}^2_2 \times \mathsf{COH}$, because both SRT^2_2 and COH have Δ^0_2-solutions to computable instances, while RT^2_2 does not (Jockusch [41]).

Another example of a problem where it is natural to consider the compositional product is the Bolzano-Weierstrass theorem.

Definition 2.26. For any computable metric space X, define BWT_X to be the following problem: given a sequence $(x_i)_i$ in X with compact closure, produce a cluster point of $(x_i)_i$.

Define WBWT_X to be the following problem: given a sequence $(x_i)_i$ in X with compact closure, produce a sequence in $\mathbb{N}^\mathbb{N}$ which converges to a name of a cluster point of $(x_i)_i$.

Example 2.27. $\mathsf{BWT}_k \equiv_W \mathsf{RT}^1_k$. Note that $\mathsf{RT}^1_k \not\leq_{sc} \mathsf{BWT}_k$: take any (noncomputable) RT^1_k-instance without any computable solutions.

It is easy to see that BWT_X and $\lim_{\mathbb{N}^\mathbb{N}} \circ \mathsf{WBWT}_X$ are the same problem. Furthermore:

Theorem 2.28 (Brattka, Hendtlass, Kreuzer [12]). *$\lim_{2^\mathbb{N}} * \mathsf{WBWT}_{2^\mathbb{N}} \leq_W \mathsf{BWT}_{2^\mathbb{N}}$. Therefore $\mathsf{BWT}_{2^\mathbb{N}} \equiv_W \lim_{2^\mathbb{N}} * \mathsf{WBWT}_{2^\mathbb{N}}$.*

Proof. Since $\lim_{2^\mathbb{N}}$ is a cylinder, by the cylindrical decomposition lemma (Lemma 2.20), there is some computable single-valued function G such that

$$\lim_{2^\mathbb{N}} * \mathsf{WBWT}_{2^\mathbb{N}} \equiv_W \lim_{2^\mathbb{N}} \circ G \circ (\mathrm{id} \times \mathsf{WBWT}_{2^\mathbb{N}}).$$

We show below that the right-hand side is strongly Weihrauch reducible to $\mathsf{J} \times \mathsf{BWT}_{2^\mathbb{N}}$, where J denotes the problem of producing the Turing jump of a given set. This completes the proof because

$$\mathsf{J} \times \mathsf{BWT}_{2^\mathbb{N}} \leq_W \mathsf{BWT}_{2^\mathbb{N}} \times \mathsf{BWT}_{2^\mathbb{N}} \leq_W \mathsf{BWT}_{2^\mathbb{N}}.$$

Take the forward functional to be the identity. For the backward functional, suppose we are given $\langle \mathsf{J}(p), q \rangle$, where q is a name for some element of $\mathsf{BWT}_{2^\mathbb{N}}(p)$. We compute a solution to $\lim_{2^\mathbb{N}}(G(p, \mathsf{WBWT}_{2^\mathbb{N}}(p)))$ as follows.

Note that any sequence in $\mathbb{N}^\mathbb{N}$ which converges to q is a solution to $\mathsf{WBWT}_{2^\mathbb{N}}(p)$. We will use $\langle \mathsf{J}(p), q \rangle$ to construct a sequence of finite strings $(u_s)_s$ such that $(u_s 0^\infty)_s$ converges to q. Along the way, we "force" $\lim_{2^\mathbb{N}}(G(p, (u_s 0^\infty)_s))$ (analogous to how one forces the jump in recursion theory).

We construct $(u_s)_s$ in stages. At stage $\langle i, k, b \rangle$ (where $i, k \in \mathbb{N}$, $b \in 2$), suppose we have constructed $u_0, \ldots, u_{s_j}$. Use $\mathsf{J}(p)$ to search for some $l > \max_{i < s_j} |u_i|$ and finitely many strings $u_{s_j+1}, \ldots, u_{s_{j+1}}$ of length l, each extending $q \upharpoonright \langle i, k, b \rangle$, such that

$$G(p, (u_0 0^{l-|u_0|}, \ldots, u_{s_j} 0^{l-|u_{s_j}|}, u_{s_j+1}, \ldots, u_{s_{j+1}}))$$

produces at least k many b's in the i^{th} row. (We think of the columns of $G(p, (u_s 0^\infty)_s)$ as binary sequences, where each row has a limit.) If such $u_{s_j+1}, \dots, u_{s_{j+1}}$ exist, then we take the least such strings and add them to our sequence. Otherwise, we do nothing. This completes the construction for stage $\langle i, k, b\rangle$.

Observe that for each i and k, we must make an extension in either stage $\langle i, k, 0\rangle$ or $\langle i, k, 1\rangle$. This is because $G(p, (u_0 0^\infty, \dots, u_{s_j} 0^\infty, q, q, \dots))$ is an instance of $\lim_{2^\mathbb{N}}$. Hence $(u_s)_s$ is an infinite sequence. Define u to be $(u_s 0^\infty)_s$.

We now show how to use $\langle \mathsf{J}(p), q\rangle$ to compute $\lim_{2^\mathbb{N}}(G(p, u))$. For each i, there must be some m and b such that we did not make an extension in stage $\langle i, m, b\rangle$. Using $\langle \mathsf{J}(p), q\rangle$, we can search for such a stage. Then $\lim_{2^\mathbb{N}}(G(p, u))(i)$ cannot be b, hence it must be $1 - b$. $\square$

We note that $\mathsf{BWT}_{2^\mathbb{N}}$ is equivalent to BWT_X for several spaces X that we care about:

Proposition 2.29 (Brattka, Gherardi, Marcone [10])**.** *If X is a computable metric space such that $2^\mathbb{N}$ computably embeds into X (e.g., $\mathbb{R}^n$ and $\mathbb{N}^\mathbb{N}$), then $\mathsf{BWT}_{2^\mathbb{N}} \equiv_{sW} \mathsf{BWT}_X$.*

Hence we will write BWT to mean $\mathsf{BWT}_{2^\mathbb{N}}$.

As for $\mathsf{WBWT}_{2^\mathbb{N}}$, it turns out to be equivalent to a familiar combinatorial principle. First, we make a definition:

Definition 2.30 ([12])**.** Define SBWT_X to be the following problem: given a sequence $(x_i)_i$ in X with compact closure, produce a convergent subsequence of $(x_i)_i$, i.e., produce $s \in \mathbb{N}^\mathbb{N}$ such that $(x_{s(n)})_n$ converges.

The proof of the following proposition is left to the reader.

Proposition 2.31 ([12])**.** $\mathsf{WBWT}_X \equiv_W \mathsf{SBWT}_X$.

Proposition 2.32. $\mathsf{SBWT}_{2^\mathbb{N}} \equiv_{sW} \mathsf{COH}$.

Proof. $\mathsf{SBWT}_{2^\mathbb{N}} \leq_{sW} \mathsf{COH}$: Given a sequence $(x_i)_i$ in $2^\mathbb{N}$, compute the following array $(R_n)_n$: for each n, $R_n = \{i : x_i(n) = 0\}$. Apply COH to obtain an infinite set C which is cohesive for $(R_i)_i$. Then if p is the principal function of C, $(x_{p(j)})_j$ converges because for each n, $x_{p(j)}(n)$ is eventually always 0 or always 1.

$\mathsf{COH} \leq_{sW} \mathsf{SBWT}_{2^\mathbb{N}}$: Given an array $(R_i)_i$, compute a sequence $(x_i)_i$ in $2^\mathbb{N}$ by $x_i(n) = R_n(i)$. Apply $\mathsf{SBWT}_{2^\mathbb{N}}$ to obtain a sequence $(s(i))_i$ in $\mathbb{N}$ such that $(x_{s(i)})_i$ converges. Let C denote the set of s_i's. Then C is cohesive for $(R_i)_i$. $\square$

Corollary 2.33 ([12])**.** $\mathsf{BWT} \equiv_W \lim * \mathsf{COH}$.

2.4. **Implication.**

Definition 2.34 (Brattka, Pauly [13])**.** Let f and g be problems. The *implication* $g \to f$ is defined to be $\min_{\leq_W}\{h : f \leq_W g * h\}$.

In other words, the implication captures the minimal Weihrauch degree that is needed in advance of g in order to compute f.

Theorem 2.35 ([13])**.** *$g \to f$ is always defined.*

Proof idea. A representative of $g \to f$ can be defined as follows. Given an f-instance u, consider the problem of producing a g-instance x and a strongly computable multivalued function Ψ, such that $\Psi(g(x)) \subseteq f(u)$. □

Remark 2.36. In the notation of the theory of residuated lattices, the implication is the *right co-residual* of the compositional product. See Brattka, Gherardi [9] for further discussion.

We give some examples of the implication.

Proposition 2.37 (Brattka, Hendtlass, Kreuzer [12])**.** $\mathsf{WBWT}_X \equiv_W \lim_{\mathbb{N}^\mathbb{N}} \to \mathsf{BWT}_X$.

Proof. We observed earlier that $\mathsf{BWT}_X = \lim \circ \mathsf{WBWT}_X \leq_W \lim * \mathsf{WBWT}_X$. On the other hand, suppose that $\mathsf{BWT}_X \leq_W \lim * h$. Without loss of generality, we may assume that h is a cylinder. By the cylindrical decomposition lemma, there is some computable function K such that $\lim * h \equiv_W \lim \circ K \circ h$. So $\mathsf{BWT}_X \leq_W \lim \circ K \circ h$, say via Γ and Δ. This allows us to show that $\mathsf{WBWT}_X \leq_W K \circ h \leq_W h$, as follows.

Given a WBWT_X-instance $(x_i)_i$, apply Γ to produce an instance of $\lim \circ K \circ h$. Suppose that $(p_n)_n$ is a $(K \circ h)$-solution to $\Gamma((x_i)_i)$. Then $(p_n)_n$ is an instance of lim. We know that $\Delta((x_i)_i, \lim((p_n)_n))$ is a name for a BWT_X-solution for $(x_i)_i$. We want to produce a sequence $(q_n)_n$ in $\mathbb{N}^\mathbb{N}$ such that $\lim(q_n)_n = \Delta((x_i)_i, \lim((p_n)_n))$. To define q_n, run $\Delta((x_i)_i, p_n)$ for n steps. Then extend its output by defining $q_n(j) = 0$ if $\Delta((x_i)_i, p_n)(j)$ has not been determined in n steps. Then $(q_n)_n$ is a WBWT_X-solution to $(x_i)_i$. □

Definition 2.38. Let $\mathsf{MLR} : 2^\mathbb{N} \rightrightarrows 2^\mathbb{N}$ denote the problem of producing a (Martin-Löf) random sequence relative to a given sequence.

Let WWKL denote the restriction of WKL to trees T such that the set of infinite paths of T has positive measure.

Proposition 2.39 ([13])**.** $\mathsf{MLR} \equiv_W \mathsf{C}_\mathbb{N} \to \mathsf{WWKL}$.

Proof. First, we show that $\mathsf{WWKL} \leq_W \mathsf{C}_\mathbb{N} * \mathsf{MLR}$. This follows from (the relativization of) a result of Kučera (see Downey, Hirschfeldt [22, Lemma 6.10.1]), which states that for any Π^0_1 class $\mathcal{C}$ of positive measure and any random A, some tail of A is an element of $\mathcal{C}$.

Proof of Kučera's lemma. Define an ML-test as follows. Let S_0 enumerate the set of leaves of a tree whose paths are exactly the elements of $\mathcal{C}$. If S_n is defined, define $S_{n+1} = \{\sigma^\frown\tau : \sigma \in S_n \wedge \tau \in S_0\}$. For each n, let U_n be the set of reals which extend some string in S_n. It is easy to check that $(U_n)_n$ can be thinned down into an ML-test. Next, since A is random, there is some n such that $A \notin U_n$. That shows that some tail of A lies in $\mathcal{C}$. (Take the largest m such that $A \in U_m$. Let $\sigma \in S_m$ be such that A extends σ. Let B be the tail of A starting from σ. Then B lies in $\mathcal{C}$ because none of its initial segments lie in S_0.) $\square$

By applying $\mathsf{C}_\mathbb{N}$, we can find a tail of A which is an element of $\mathcal{C}$.

Second, suppose that $\mathsf{WWKL} \le_W \mathsf{C}_\mathbb{N} * h$ for some h. Without loss of generality, we may assume that h is a cylinder, so $\mathsf{WWKL} \le_W f \circ g$ for some $f \le_W \mathsf{C}_\mathbb{N}$ and $g \le_{sW} h$. We show that $\mathsf{MLR} \le_W g$ (hence $\mathsf{MLR} \le_W h$ as desired). The point is that MLR is Weihrauch reducible to WWKL with finite error, and $\mathsf{C}_\mathbb{N}$ can be computed with finitely many mind-changes.

Given X, let $(U_i^X)_i$ be a universal Martin-Löf test relative to X. (There is a single index defining $(U_i^X)_i$ which works for all X.) Consider the complement $\mathcal{C}$ of U_0^X. It is a $\Pi^{0,X}_1$ class of positive measure, and every element of $\mathcal{C}$ is random. Transform $\mathcal{C}$ into an $(f \circ g)$-instance. This completes the definition of the forward functional.

Suppose we apply g to obtain a g-solution X, which is itself an f-instance. By assumption, X uniformly enumerates a proper subset $S \subseteq \mathbb{N}$, such that given any $s \notin S$, we can uniformly (in $\mathcal{C} \oplus X$) compute an element of $\mathcal{C}$. Instead of appealing to $\mathsf{C}_\mathbb{N}$, we guess a $\mathsf{C}_\mathbb{N}$-solution and attempt to compute an element of $\mathcal{C}$, changing our guess whenever it is proven wrong by the X-computable enumeration. Since we only change our guess finitely many times, we end up producing some q which differs from an element of $\mathcal{C}$ on an initial segment (at most). But then q differs with a random on an initial segment, hence it is random as well. $\square$

Note that WWKL is *not* Weihrauch equivalent to $\mathsf{C}_\mathbb{N} * \mathsf{MLR}$ because $\mathsf{C}_\mathbb{N} \not\le_W \mathsf{WWKL}$ (or even WKL, as shown by Brattka, de Brecht, Pauly [6, Corollary 5.2]).

Another example of an implication involving $\mathsf{C}_\mathbb{N}$ is:

Proposition 2.40 (Dzhafarov, Goh, Hirschfeldt, Patey, Pauly [26]). *RT^2_2 with finite error (i.e., solutions agree with some infinite homogeneous set on a cofinite set) is a representative of the degree $\mathsf{C}_\mathbb{N} \to \mathsf{RT}^2_2$.*

2.5. **Jumps.**

Definition 2.41 (Brattka, Gherardi, Marcone [10])**.** The *jump* of a represented space (X, δ) is defined to be (X, δ'), where $\delta' = \delta \circ \lim$.

The *jump* of a problem $f :\subseteq (X, \delta_X) \rightrightarrows (Y, \delta_Y)$, denoted f', is defined to be $f :\subseteq (X, \delta'_X) \rightrightarrows (Y, \delta_Y)$.

In other words, f' is the following problem: given a sequence in $\mathbb{N}^{\mathbb{N}}$ which converges to a name of an f-instance, solve said f-instance.

Example 2.42. $\mathrm{id}' \equiv_{sW} \lim$.

We always have $f \leq_{sW} f'$, but we need not have $f <_{sW} f'$.

Example 2.43. If f is a pointed constant function, then $f \equiv_{sW} f'$.

It follows that the jump is *not* monotone with respect to Weihrauch reducibility: take f to be any computable pointed constant function. Then $\mathrm{id} \leq_W f$ but $\mathrm{id}' \equiv_{sW} \lim \not\leq_W f \equiv_{sW} f'$.

Nevertheless, the jump is monotone with respect to strong Weihrauch reducibility. First, we need a lemma:

Lemma 2.44. *Given any computable single-valued function* $\Delta :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ *and any convergent sequence* $(p_n)_n$ *in* $\mathbb{N}^{\mathbb{N}}$ *such that* $\lim((p_n)_n) \in \mathrm{dom}(\Delta)$*, we can uniformly compute some sequence* $(q_n)_n$ *in* $\mathbb{N}^{\mathbb{N}}$ *such that* $\lim((q_n)_n) = \Delta(\lim((p_n)_n))$*.*

The proof of the lemma is similar to that of Proposition 2.37 (indeed, we should have proved this lemma earlier and used it to prove said proposition).

Proposition 2.45 ([10])**.** *If* $f \leq_{sW} g$*, then* $f' \leq_{sW} g'$*.*

Proof. Fix Γ and Δ witnessing that $f \leq_{sW} g$. Given a sequence $(p_n)_n$ in $\mathbb{N}^{\mathbb{N}}$ which converges to a name for an f-instance X, we want to compute a sequence $(q_n)_n$ in $\mathbb{N}^{\mathbb{N}}$ which converges to a name for the g-instance $\Gamma(X)$. This can be done by Lemma 2.44. This completes the definition of the forward functional. Take the backward functional to be Δ. □

Observe that the above proof fails for ordinary Weihrauch reducibility because in that case, in order for the backward functional to utilize Δ, it has to first produce $\lim((p_n)_n)$.

The above proposition exhibits the utility of strong Weihrauch reducibility as a technical tool; strong Weihrauch reductions between problems yield strong Weihrauch reductions between their jumps.

Next, observe that $f' \leq_W f * \lim$. If f is a cylinder, then the converse holds.

Proposition 2.46 ([10]). *If f is a cylinder, then f' is a cylinder and $f' \equiv_W f' \times \lim \equiv_W f * \lim$.*

Proof. If $\mathrm{id} \times f \leq_{sW} f$, then

$$\mathrm{id} \times f' \leq_{sW} \mathrm{id}' \times f' \equiv_{sW} \lim \times f' \leq_{sW} (\mathrm{id} \times f)' \leq_{sW} f',$$

so f' is a cylinder. Since $f' \leq_{sW} \mathrm{id} \times f'$, it also follows that $f' \equiv_{sW} f' \times \lim$.

It remains to show that $f * \lim \leq_W f'$. By the cylindrical decomposition lemma, there is some computable single-valued function Φ such that $f * \lim \equiv_W f \circ \Phi \circ \lim$. We show that $f \circ \Phi \circ \lim \leq_{sW} f'$. For the forward functional, given an instance $(p_n)_n$ of $f \circ \Phi \circ \lim$, we can uniformly compute a sequence $(q_n)_n$ in $\mathbb{N}^\mathbb{N}$ which converges to a name for $\Phi(\lim((p_n)_n))$ (Lemma 2.44). For the backward functional, use the identity. □

Corollary 2.47. *If g is a cylinder, then $f \leq_W g$ implies $f' \leq_{sW} g'$. Therefore for any g, $(g \times \mathrm{id})' \equiv_W \max_{\leq_W}\{f' : f \leq_W g\}$.*

This would allow us to lift the jump operation to Weihrauch degrees. But we will not do so; in particular, we will still use f' to denote the jump of f, rather than the jump of $\mathrm{id} \times f$.

For problems which may not be cylinders, we still have the following:

Proposition 2.48 ([10]). *For any problem f,*

$$\max_{\leq_{sW}}\{f_0 \circ g_0 : f_0 \leq_{sW} f, g_0 \leq_{sW} \lim\}$$

exists and is strongly Weihrauch equivalent to f'.

Proof. Let $f^r :\subseteq \mathbb{N}^\mathbb{N} \rightrightarrows \mathbb{N}^\mathbb{N}$ be the realizer version of f, i.e., $f^r = \delta_Y \circ f \circ \delta_X^{-1}$. Then $f^r \equiv_{sW} f$ and $f' \equiv_{sW} f^r \circ \lim$.

Conversely, it suffices (by the usual arguments) to show that for any computable single-valued function Φ, $f \circ \Phi \circ \lim \leq_{sW} f'$. Proceed as in the proof of Proposition 2.46. □

Next, we present some examples of jumps among the problems that we have discussed in this course.

Proposition 2.49. $\mathsf{C}_k' \equiv_{sW} \mathsf{BWT}_k$. *(Recall $\mathsf{BWT}_k \equiv_W \mathsf{RT}^1_k$.)*

Proof. $\mathsf{BWT}_k \leq_{sW} \mathsf{C}_k'$: Consider the problem f, defined as follows: given a sequence $(x_i)_i$ in $k^\mathbb{N}$, enumerate every number which is not a cluster point of $(x_i)_i$, i.e., enumerate every number below k which appears at most finitely many times in $(x_i)_i$. Observe that $\mathsf{BWT}_k \leq_{sW} \mathsf{C}_k \circ f$. By Proposition 2.48, if we show that $f \leq_{sW} \lim$, it would follow that $\mathsf{BWT}_k \leq_{sW} \mathsf{C}_k'$ as desired.

To reduce f to lim, use lim to compute, for each $j < k$ and each $n \in \mathbb{N}$, whether j appears in $(x_i)_{i>n}$. If there is some $n \in \mathbb{N}$ such that j does not appear in $(x_i)_{i>n}$, then we enumerate j.

$\mathsf{C}_k' \leq_{sW} \mathsf{BWT}_k$: Suppose we are given a sequence $(p_n)_n$ in $\mathbb{N}^\mathbb{N}$ which converges to a name for an enumeration of a proper subset $A \subsetneq k$. (For concreteness, the enumeration named by $p \in \mathbb{N}^\mathbb{N}$ enumerates $p(s)$ at stage s if $p(s) < k$, otherwise it does not enumerate any number.) We want to compute a sequence $(x_i)_i$ in $k^\mathbb{N}$ whose cluster points are exactly the numbers not in A, i.e., every number in A occurs at most finitely often, while every number outside A occurs infinitely often.

We define $(x_i)_i$ by checking for each s and j whether the following Σ^0_1 fact holds:

$$(\exists n > s)(p_n \text{ has not enumerated } j \text{ by stage } s).$$

Whenever we realize that the above holds for (j, s, n), we put j in the sequence we are defining.

If j lies in A, then there is some t and m such that for all $n > m$, p_n enumerates j at stage t. Hence the above Σ^0_1 fact must fail for all $s > \max\{t, m\}$. So j occurs at most finitely often in our sequence.

On the other hand, if j does not lie in A, then for all s, there is some $n > s$ such that p_n has not enumerated j by stage s. (Otherwise, by the infinite pigeonhole principle, there would be some $t < s$ such that for infinitely many $n > s$, p_n enumerates j at stage t. Since $(p_n)_n$ converges, this implies that j is enumerated in the limit.) So j occurs infinitely often in our sequence. □

Note that the above proposition is a special case of the following:

Theorem 2.50 ([10, Theorem 9.4])**.** *C_X' is strongly Weihrauch equivalent to the problem of producing a cluster point of a sequence (with domain being the set of sequences which have some cluster point).*

Corollary 2.51. *$\mathsf{BWT} \equiv_{sW} \mathsf{WKL}'$.*

Proof. It is easy to see that $\mathsf{WKL} \equiv_{sW} \mathsf{C}_{2^\mathbb{N}}$. By Proposition 2.45, $\mathsf{WKL}' \equiv_{sW} \mathsf{C}_{2^\mathbb{N}}'$. Next, by Theorem 2.50, $\mathsf{C}_{2^\mathbb{N}}' \equiv_{sW} \mathsf{BWT}_{2^\mathbb{N}}$. (The cluster point problem for $2^\mathbb{N}$ is equivalent to $\mathsf{BWT}_{2^\mathbb{N}}$ because $2^\mathbb{N}$ is compact.) □

Proposition 2.52. *$\mathsf{WKL}' \equiv_{sW} \widehat{\mathsf{BWT}_2}$.*

Proof. We showed earlier that $\mathsf{WKL} \equiv_{sW} \widehat{\mathsf{C}_2}$ (Proposition 2.11). By Proposition 2.45, $\mathsf{WKL}' \equiv_{sW} \left(\widehat{\mathsf{C}_2}\right)'$. It is easy to see that parallelization commutes with jumps, so $\mathsf{WKL}' \equiv_{sW} \widehat{\mathsf{C}_2'}$. Finally, we showed above that $\mathsf{C}_2' \equiv_{sW} \mathsf{BWT}_2$, so $\widehat{\mathsf{C}_2'} \equiv_{sW} \widehat{\mathsf{BWT}_2}$. ☐

For computable metric spaces X which are not necessarily compact, BWT_X is known to be equivalent to the jump of the *compact choice* problem, defined as follows. If X is a computable metric space, we represent the set of compact subsets of X as follows: p is a name for a compact set $K \subseteq X$ if p enumerates all tuples which code a finite cover of K by rational open balls.

Observe that given a name for K as a compact set, we can compute a name for K as a closed set. This follows from the following two facts. First, given any finite collection of rational open balls, we can enumerate all rational open balls which are disjoint from their union. Second, any element not in K is contained in some open ball whose closure is disjoint from K. By compactness of K, any such open ball is disjoint from some finite open cover of K.

Definition 2.53. Define the problem of *compact choice* in a computable metric space X, written K_X, as follows: given a compact set (represented as above), choose an element of the compact set.

Proposition 2.54. *For any computable metric space X, $\mathsf{K}_X \leq_{sW} \mathsf{C}_X$.*

Proof. For the forward functional, see the paragraph above Definition 2.53. Take the backward functional to be the identity. ☐

In general, K_X is weaker than C_X.

Example 2.55 ([10, Corollary 10.10])**.** $\mathsf{K}_{\mathbb{N}} \equiv_{sW} \mathsf{C}_2^* <_W \mathsf{C}_{\mathbb{N}}$.

Theorem 2.56 ([10, Theorem 11.2])**.** $\mathsf{BWT}_X \equiv_{sW} \mathsf{K}_X'$.

Finally, we show that BWT is equivalent to König's lemma KL.

Definition 2.57. Let KL denote the following problem: given an infinite finitely branching subtree of $\mathbb{N}^{<\mathbb{N}}$, produce an infinite path.

Theorem 2.58. $\mathsf{KL} \equiv_{sW} \mathsf{BWT} \equiv_{sW} \mathsf{WKL}' \equiv_{sW} \widehat{\mathsf{BWT}_2}$.

Proof. First, we show that $\mathsf{KL} \leq_{sW} \mathsf{BWT}_{\mathbb{N}^{\mathbb{N}}}$. Given an infinite finitely branching tree $T = \{\sigma_0, \sigma_1, \dots\}$, consider the sequence $(\sigma_i 0^\infty)_i$ in $\mathbb{N}^{\mathbb{N}}$. Since T is finitely branching, $(\sigma_i 0^\infty)_i$ has compact closure. If p is a cluster point of $(\sigma_i 0^\infty)_i$, that means that for each n, there are infinitely many i such that $p \upharpoonright n$ is an initial segment of $\sigma_i 0^\infty$. Since there are only finitely many strings in T of length shorter than n, it follows that there is some i such that $p \upharpoonright n$ is an initial segment of σ_i. So $p \upharpoonright n$ lies in T. Therefore p is an infinite path on T.

Next, Corollary 2.51 states that $\mathsf{BWT} \equiv_{sW} \mathsf{WKL}'$. Proposition 2.52 states that $\mathsf{WKL}' \equiv_{sW} \widehat{\mathsf{BWT}_2}$.

To complete the proof, we show that $\widehat{\mathsf{BWT}_2} \leq_{sW} \widehat{\mathsf{KL}} \leq_{sW} \mathsf{KL}$. First, we show that $\mathsf{BWT}_2 \leq_{sW} \mathsf{KL}$ (and hence $\widehat{\mathsf{BWT}_2} \leq_{sW} \widehat{\mathsf{KL}}$). Given a binary sequence $(b_i)_i$, define an infinite finitely branching tree T as follows. For each $n \geq 1$ and $b < 2$, define the string σ_n^b of length n as follows: $\sigma_n^b(0) = b$, and $\sigma_n^b(1) < \sigma_n^b(2) < \cdots < \sigma_n^b(n-1)$ lists the first $(n-1)$-many i's such that $b_i = b$. Let $T = \{\sigma_n^b : n \geq 1, b < 2\}$.

Since T contains at most two strings at each level, it is finitely branching. It is clear that T is infinite, and if p is an infinite path on T, then $p(0)$ appears infinitely many times in $(b_i)_i$. This proves that $\mathsf{BWT}_2 \leq_{sW} \mathsf{KL}$.

The proof that $\widehat{\mathsf{KL}} \leq_{sW} \mathsf{KL}$ is the same as the proof that $\widehat{\mathsf{WKL}} \leq_{sW} \mathsf{WKL}$. □

See Brattka, Rakotoniaina [15, Theorem 5.13] for a direct reduction from $\mathsf{BWT}_{\mathbb{N}^\mathbb{N}}$ to KL.

Remark 2.59. In reverse mathematics, König's lemma and the Bolzano-Weierstrass theorem are both equivalent to ACA_0 over the standard base theory RCA_0.

Corollary 2.60 ([15, Corollary 5.14])**.** *$\mathsf{KL} \leq_W \mathsf{RT}^3_2$.*

Proof. First, $\mathsf{KL} \equiv_{sW} \widehat{\mathsf{BWT}_2}$ (Theorem 2.58). Since $\mathsf{BWT}_2 \equiv_W \mathsf{RT}^1_2$, it follows that $\mathsf{KL} \equiv_W \widehat{\mathsf{RT}^1_2}$. We show that for any k, $\widehat{\mathsf{RT}^1_k} \leq_{sW} \mathsf{RT}^3_2$. Given a sequence $(c_i)_i$ of colorings $c_i : \mathbb{N} \to k$, define a coloring $c : [\mathbb{N}]^3 \to 2$ by

$$c(m, x, y) = \begin{cases} 0 & \text{if } (\forall i < m)[c_i(x) = c_i(y)] \\ 1 & \text{otherwise} \end{cases}.$$

Suppose that H is an infinite c-homogeneous set. We claim that the c-color of H must be 0. Let $m = \min(H)$. If H has color 1, we can define a coloring $d : [H]^2 \to m$ by taking $d(x, y)$ to be the least $i < m$ such that $c_i(x) \neq c_i(y)$. By Ramsey's theorem for pairs, there is some infinite d-homogeneous set $H' \subseteq H$. But the range of each c_i is at most k, so d cannot have a homogeneous set of size greater than k. Contradiction. This proves our claim.

Finally, for each i, we compute an infinite c_i-homogeneous set as follows. Let m be the least number in H above i. Since $H \backslash [0, m)$ is c-homogeneous, $H \backslash [0, m]$ is c_i-homogeneous. □

The above result was obtained independently by Hirschfeldt, Jockusch [36, Corollary 2.3]. Their proof does not involve BWT. Instead, given an X-computable infinite finitely branching tree, they construct an X-computable 2-coloring of triples such that if H is an infinite homogeneous set, then $X \oplus H$ has PA degree over $\emptyset'$.

Next, we turn to stable Ramsey's theorem. We would like to say that $\mathsf{SRT}^2_k \equiv_W (\mathsf{RT}^1_k)'$. It is true that $(\mathsf{RT}^1_k)' \leq_{sW} \mathsf{SRT}^2_k$, but Dzhafarov [24, Corollary 3.3] showed that $\mathsf{SRT}^2_2 \not\leq_W (\mathsf{RT}^1_k)'$.

Definition 2.61 (Brattka, Rakotoniaina [15])**.** CRT^n_k is defined by enriching the output of RT^n_k with the color of the homogeneous set in the output, i.e., given a coloring $c : [\mathbb{N}]^n \to k$, output an infinite c-homogeneous set and its color.

Trivially $\mathsf{RT}^n_k \leq_{sW} \mathsf{CRT}^n_k$ and $\mathsf{CRT}^n_k \leq_W \mathsf{RT}^n_k$. However, CRT^n_k is not strongly Weihrauch reducible to RT^n_k (or even $(\mathsf{RT}^n_k)'$). Intuitively, this is because any finite number of RT^n_k-instances have a common solution, so one cannot uniformly extract much information from an RT^n_k-solution. See [15, Corollary 3.15] for details.

Theorem 2.62 (Brattka, Rakotoniaina [15, Theorem 4.3])**.** $\mathsf{SRT}^2_k \equiv_W (\mathsf{CRT}^1_k)'$.

Proof. $(\mathsf{CRT}^1_k)' \leq_W \mathsf{SRT}^2_k$: Suppose we are given a sequence $(c_i)_i$ which converges to a coloring $c_\infty : \mathbb{N} \to k$. By adjusting the c_i's, we may assume that each c_i is a coloring $c_i : \mathbb{N} \to k$ as well.

Define a coloring $c : [\mathbb{N}]^2 \to k$ as follows: $c(x, i) = c_i(x)$. c is stable because for all x, $\lim_i c_i(x)$ exists. Any infinite c-homogeneous set is also c_∞-homogeneous with the same color. (This is not a strong Weihrauch reduction because we use our access to c to determine the color of the homogeneous set. But the above proof shows that $(\mathsf{RT}^1_k)' \leq_{sW} \mathsf{SRT}^2_k$.)

$\mathsf{SRT}^2_k \leq_W (\mathsf{CRT}^1_k)'$: Given a stable coloring $c : [\mathbb{N}]^2 \to k$, we define a sequence of colorings $c_n : \mathbb{N} \to k$ as follows:

$$c_n(x) = \begin{cases} c(x, n) & x < n \\ 0 & \text{otherwise} \end{cases}.$$

The sequence $(c_n)_n$ converges because for all x, $\lim_n c(x, n)$ exists. Denote its limit by $c_\infty : \mathbb{N} \to k$. Now given any infinite c_∞-homogeneous set H of color j, we may thin it out (using both the coloring c and the color j!) to obtain an infinite c-homogeneous set. □

2.6. **Other algebraic properties.**

Definition 2.63. A *mass problem* is a subset of $\mathbb{N}^\mathbb{N}$. A mass problem A is *Medvedev reducible* to a mass problem B, written $A \leq_M B$, if given any element of B, one can uniformly compute an element of A.

For any mass problems A and B, their join $A + B$ is defined to be $\{a \oplus b : a \in A, b \in B\}$. Their meet $A \times B$ is defined to be the disjoint union of A and B, i.e., $\{0^\frown a : a \in A\} \cup \{1^\frown b : b \in B\}$.

Medvedev reducibility induces a degree structure on mass problems. The join and meet lift to the Medvedev degrees. The Medvedev degrees form a distributive lattice with minimum element $\mathbb{N}^\mathbb{N}$ and maximum element $\emptyset$.

There are two ways to embed the Medvedev degrees into the Weihrauch degrees:

– map nonempty $A \subseteq \mathbb{N}^\mathbb{N}$ to the problem c_A of producing an element of A, given an arbitrary element of $\mathbb{N}^\mathbb{N}$ ([8]);
– map $A \subseteq \mathbb{N}^\mathbb{N}$ to the problem d_A: given an element of A, produce 0 ([34]).

The first embedding is order-preserving and meet-preserving. However, $c_{A+B} \equiv_W c_A \times c_B$ rather than $c_A \sqcup c_B$, so this is not a lattice embedding.

The second embedding reverse-embeds the Medvedev lattice into the Weihrauch lattice, i.e.,

– $A \leq_M B$ if and only if $d_B \leq_W d_A$;
– $d_{A+B} \equiv_W d_A \sqcap d_B$;
– $d_{A\times B} \equiv_W d_A \sqcup d_B$.

Higuchi, Pauly [34] observed that the image of $A \mapsto d_A$ in the Weihrauch degrees is exactly the cone below id. As for $A \mapsto c_A$:

Proposition 2.64 (Brattka, Pauly [13, §5]). *The image of $A \mapsto c_A$ in the Weihrauch degrees is exactly every degree of the form $\mathbf{a} \to \mathrm{id}$. Moreover, for each A, $c_A \equiv_W d_A \to \mathrm{id}$.*

The many-one semilattice can be embedded into the Weihrauch lattice as well (Brattka, Gherardi, Pauly [11, Proposition 9.2]). Fix Turing-incomparable $p, q \in \mathbb{N}^\mathbb{N}$. For each $A \subseteq \mathbb{N}$, define $m_A : \mathbb{N} \to \mathbb{N}^\mathbb{N}$ by

$$m_A(n) = \begin{cases} p & \text{if } n \in A \\ q & \text{if } n \notin A \end{cases}.$$

This is a join-semilattice embedding from the many-one semilattice into the Weihrauch lattice.

Proposition 2.65 (Higuchi, Pauly [34, Proposition 3.15]). *The Weihrauch lattice has no nontrivial countable suprema, i.e., if $\{f_n : n \in \mathbb{N}\}$ has a supremum g, then $g \leq_W \bigsqcup_{i<n} f_i$ for some n. Equivalently, if $f_0 <_W f_1 <_W \dots$, then $\sup_{\leq_W}\{f_n : n \in \mathbb{N}\}$ does not exist.*

Proof. Suppose that $g \equiv_W \sup_{\leq_W}\{f_n : n \in \mathbb{N}\}$. Without loss of generality, we may assume that g and each f_n are (possibly partial) multivalued functions on $\mathbb{N}^\mathbb{N}$.

We will construct some $h :\subseteq \mathbb{N}^\mathbb{N} \rightrightarrows \mathbb{N}^\mathbb{N}$ such that $f_n \leq_W h$ for every n. In order to do so, we will construct an appropriate sequence $(a_n)_n$ in $\mathbb{N}$ and define $h(a_n{}^\frown p) = f_n(p)$ for each $p \in \mathrm{dom}(f_n)$.

We construct $(a_n)_n$ by "diagonalizing" against every possible forward reduction from g to h. At stage n, define a_n as follows. Let Φ_n denote the n^{th} Turing functional. If there is some $p \in \mathrm{dom}(g)$ such that $\Phi_n(p)(0)$ converges and is greater than a_{n-1}, then take any such p and define $a_n = \Phi_n(p)(0) + 1$. Otherwise, define $a_n = a_{n-1} + 1$.

Now, since $f_n \leq_W h$ for every n, we have that $g \leq_W h$. Suppose that Φ_n and Ψ witness that $g \leq_W h$. In particular, for all $p \in \mathrm{dom}(g)$, $\Phi_n(p)(0)$ converges. If there is some $p \in \mathrm{dom}(g)$ such that $\Phi_n(p)(0)$ is greater than a_{n-1}, we would have ensured that $\Phi_n(p) \notin \mathrm{dom}(h)$, contradiction. Hence for all $p \in \mathrm{dom}(g)$, $\Phi_n(p)(0) \leq a_{n-1}$. This implies that $g \leq_W \bigsqcup_{i \leq a_{n-1}} f_i$ via Φ_n and Ψ. □

As for infima, the analogous result holds if we restrict ourselves to the pointed Weihrauch degrees:

Proposition 2.66 ([34, Corollary 3.18])**.** *The pointed Weihrauch lattice has no nontrivial countable infima, i.e., if $f_0 >_W f_1 >_W \dots$ are pointed, then $\inf_{\leq_W}\{f_n : n \in \mathbb{N}\}$ does not exist.*

Proof. Suppose that $g \leq_W f_n$ for every n. Without loss of generality, we may assume that g and each f_n are (possibly partial) multivalued functions on $\mathbb{N}^\mathbb{N}$. For each $m < n$, fix $\Gamma_{n,m}$ and $\Delta_{n,m}$ witnessing that $f_n \leq_W f_m$.

For each i, j, we will construct some problem $h_{\langle i,j\rangle}$ such that if Φ_i and Φ_j witness that $h_{\langle i,j\rangle} \leq_W g$, then $f_{\langle i,j\rangle} \leq_W g$ (which is a contradiction). We will also construct an auxiliary sequence $(a_{\langle i,j\rangle})_{i,j}$ in $\mathbb{N}$. Then we will define $h = \bigcup_n h_n$.

At stage $n = \langle i, j\rangle$, we construct h_n and a_n as follows. Suppose that in earlier stages we have defined $a_0, \dots, a_{m-1}$. In order to motivate the definition of a_n, we begin by presenting the definition of h_n, assuming that a_n has been defined. An instance of h_n is a tuple

$$\langle n, \Gamma_{n,0}(p), \dots, \Gamma_{n,n-1}(p), p\rangle,$$

where p is an instance of f_n. For ease of notation, we denote the above tuple by $\alpha(n, p)$. This is purely notational; note that $\alpha(n, p)$ cannot, in general, be computed uniformly from n and p.

An h_n-solution to the above tuple is some $a{}^\frown q \in \mathbb{N}^\mathbb{N}$ which satisfies one of the following conditions:

(1) $a = a_m$ for some $m < n$ and q is an f_m-solution to $\Gamma_{n,m}(p)$;
(2) $a = a_n$ and q is an f_n-solution to p;
(3) $a > a_n$ and $q = 0^{\mathbb{N}}$.

Condition (3) will be useful later, for showing that $h \leq_W f_m$ for every m. This completes the definition of h_n, assuming $a_0, \dots, a_n$ have been defined.

a_n is defined as follows. If there is some f_n-instance p such that $\Phi_i(\alpha(n,p))$ is a g-instance, and some q which is a g-solution to $\Phi_i(\alpha(n,p))$ such that $\Phi_j(\alpha(n,p),q)(0) > a_{n-1}$, then we define

$$a_n = \Phi_j(\alpha(n,p),q)(0) + 1.$$

Otherwise, define $a_n = a_{n-1} + 1$.

We claim that if $n = \langle i, j\rangle$ and Φ_i and Φ_j witness that $h_n \leq_W g$, then $f_n \leq_W g$. First note that for every f_n-instance p, $\Phi_i(\alpha(n,p))$ is a g-instance. If there is some f_n-instance p and some g-solution q to $\Phi_i(\alpha(n,p))$ such that $\Phi_j(\alpha(n,p),q)(0) > a_{n-1}$, then by definition of a_n,

$$a_{n-1} < \Phi_j(\alpha(n,p),q)(0) < a_n.$$

By definition of h_n, there is no h_n-solution to $\alpha(n,p)$ which begins with $\Phi_j(\alpha(n,p),q)$. This contradicts our assumption that Φ_i and Φ_j witness that $h_n \leq_W g$.

Therefore, for every f_n-instance p and g-solution q to $\Phi_i(\alpha(n,p))$, we have that $\Phi_j(\alpha(n,p),q)(0) \leq a_{n-1}$. By definition of h_n, this means that $\Phi_j(\alpha(n,p),q)$ must be of the form $a_m{}^\frown r$, where $m < n$ and r is an f_m-solution to $\Gamma_{n,m}(p)$.

This allows us to reduce f_n to g, as follows. Our reduction will use the following finite information for each $m < n$: a_m, $\Gamma_{n,m}$, and $\Delta_{n,m}$. Given an f_n-instance p, compute the g-instance $\Phi_i(\alpha(n,p))$. This can be done uniformly using the above finite information.

Given a g-solution q to $\Phi_i(\alpha(n,p))$, let $\Phi_j(\alpha(n,p),q) = a^\frown r$. We can check a against our list $a_0, \dots, a_{n-1}$ to find $m < n$ such that $a = a_m$. Then r is an f_m-solution to $\Gamma_{n,m}(p)$. So $\Delta_{n,m}(p,r)$ is an f_n-solution to p. This completes the proof of our claim.

Next, define $h = \bigcup_n h_n$. It follows from our claim that $h \not\leq_W g$. It remains to show that for every m, $h \leq_W f_m$. Fix m. Since f_m is pointed, we may fix a computable f_m-instance c. Our reduction from h to f_m will use the finite information $a_0, \dots, a_m$.

Suppose we are given an h-instance $\alpha(n,p)$. If $n < m$, we compute the f_m-instance c which we fixed above. Apply f_m to c. By condition (3) in our definition of h_n, $(a_n + 1)^\frown 0^{\mathbb{N}}$ is an h-solution to $\alpha(n,p)$.

If $n = m$, then apply f_m to p to obtain some f_m-solution q. Then by condition (2) in our definition of h_n, $a_m{}^\frown q$ is an h-solution to $\alpha(n,p)$.

Finally, if $n > m$, then apply f_m to $\Gamma_{n,m}(p)$ to obtain some f_m-solution q. (This is why we included $\Gamma_{n,m}(p)$ in $\alpha(n,p)$!) Then by condition (1) in our definition of h_n, $a_m{}^\frown q$ is an h-solution to $\alpha(n,p)$. This proves that $h \leq_W f_m$, as desired. $\square$

We mention that the Weihrauch lattice (specifically the cone below id) does have nontrivial countable infima. This is because there are nontrivial countable suprema in the Medvedev lattice, which is reverse-isomorphic to the cone below id in the Weihrauch lattice.

3. Some hyperarithmetic theory

The goal of this section is to present enough hyperarithmetic theory for the reader to follow the arguments in subsequent sections. For an introduction to hyperarithmetic theory, the following references may be helpful: Sacks [51], Ash, Knight [3], Chong, Yu [18].

Definition 3.1. Let L be a linear ordering with first element 0_L, and let $A \subseteq \mathbb{N}$. We say that $\langle X_a \rangle_{a \in L}$ is a *jump hierarchy on L which starts with A* if $X_0 = A$ and for all $b >_L 0_L$, $X_b = (\bigoplus_{a <_L b} X_a)'$. If we do not specify the starting set of a jump hierarchy, we assume that it is $\emptyset$.

We say that $A \subseteq \mathbb{N}$ is *B-hyperarithmetic*, or A is *hyperarithmetically reducible* to B, written $A \leq_h B$, if A is computable in some jump hierarchy on some B-computable well-ordering L which starts with B. The class of all B-hyperarithmetic sets is denoted $\mathrm{HYP}(B)$. For B computable, we simply denote it by HYP.

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

The B-hyperarithmetic sets can also be characterized as the class of subsets of $\mathbb{N}$ which are definable by some B-computable infinitary formula (see [3, Chapter 7]).

The least ordinal which is not the ordertype of a B-computable well-ordering, is denoted ω_1^B. For B computable, we denote it by ω_1^{CK} (Church-Kleene). We can define the α^{th} jump of B for each $\alpha < \omega_1^B$, which by work of Spector is canonical up to Turing degree. Therefore the B-hyperarithmetical sets are stratified by the ordertypes of B-computable well-orderings.

The most important technique in hyperarithmetical theory is effective transfinite recursion. We start with the recursion theorem:

Theorem 3.2. *If $F : \mathbb{N} \to \mathbb{N}$ is total X-computable, then there is some $e \in \mathbb{N}$ such that $\varphi_e^X = \varphi_{F(e)}^X$, i.e., φ_e^X and $\varphi_{F(e)}^X$ have the same domain* *and agree on said domain. Furthermore, we can compute some such e from an index of F as an X-computable function, which satisfies the above property for all X.*

Then we present effective transfinite recursion:

Theorem 3.3. *Let L be a linear ordering. Suppose that $F : \mathbb{N} \to \mathbb{N}$ is a total X-computable function such that for any $e \in \mathbb{N}$ and any $b \in L$,*

$$\big(\forall a <_L b[\varphi_e^X(a)\downarrow]\big) \to \varphi_{F(e)}^X(b)\downarrow .$$

Then there is some $e \in \mathbb{N}$ such that $\varphi_e^X = \varphi_{F(e)}^X$ and $\{b \in L : \varphi_e^X(b)\uparrow\}$ is either empty or contains an infinite $<_L$-descending sequence. Furthermore, we can compute some such e from an index of F as an X-computable function, which satisfies the above property for all X.

Proof. By the recursion theorem, compute some e such that $\varphi_e^X = \varphi_{F(e)}^X$. If $\{b \in L : \varphi_e^X(b)\uparrow\}$ is nonempty, then it cannot have an $<_L$-least element because for any $b \in L$,

$$\big(\forall a <_L b[\varphi_e^X(a)\downarrow]\big) \to \varphi_e^X(b)\downarrow .$$

$\square$

Observe that effective transfinite recursion does not require L to be effective in any way.

To illustrate effective transfinite recursion, we prove that every jump hierarchy on a well-ordering is a Π^0_2-singleton (relative to appropriate parameters).

Theorem 3.4. *Let L be a well-ordering, and let $X = \langle X_a\rangle_{a\in L}$ be a jump hierarchy on L which starts with A. Then each X_a is a $\Pi_2^{0,L\oplus A}$-singleton, i.e., there is a $\Pi_2^{0,L\oplus A}$ predicate $P(Y)$ such that $P(Y)$ holds if and only if $Y = X_a$.*

Let P_e denote the e^{th} Π^0_2 predicate. We prove the theorem assuming that L and A are computable. The proof of the full theorem follows by relativization.

Proposition 3.5. *If X is a Π^0_2-singleton, then so is X'. Furthermore, we can compute an index for X' as a Π^0_2-singleton from an index for X as a Π^0_2-singleton.*

Proof. Fix an index e such that $\Phi_e^{Z'} = Z$ for any Z, and Φ_e^Y is total for all Y. Then X' is the unique set Y which satisfies the following Π^0_2 formula:

$$\Phi_e^Y = X \text{ and } \forall n(n \in Y \leftrightarrow \Phi_n^{\Phi_e^Y}(n)\downarrow).$$

$\square$

Proof of Theorem 3.4. We proceed by effective transfinite recursion along L. Define a total computable function $F : \mathbb{N} \to \mathbb{N}$ as follows. Given d, we define $F(d)$ by defining $\varphi_{F(d)}(b)$ for each b.

If $b \notin L$, let $\varphi_{F(d)}(b)$ diverge. If b is the least element of L, let $\varphi_{F(d)}(b)$ be an index for the Π^0_2 formula $Y = A$. Otherwise, let e be an index for the following Π^0_2 formula $P(Y)$:

$$\forall a(\text{if } (a <_L b \wedge \varphi_d(a){\downarrow}) \text{ then } P_{\varphi_d(a)}(Y^{[a]}), \text{ otherwise } Y^{[a]} = \emptyset)$$

Then define $\varphi_{F(d)}(b) = h(e)$, where h is a total computable function such that if e is an index for a Π^0_2-singleton X, then $h(e)$ is an index for X'.

Observe that for any $d \in \mathbb{N}$ and any $b \in L$, $\varphi_{F(d)}(b){\downarrow}$. By effective transfinite recursion, we obtain some $d \in \mathbb{N}$ such that $\varphi_d = \varphi_{F(d)}$. Then $\varphi_d(b){\downarrow}$ for all $b \in L$. Finally, by transfinite induction along L, we can show that for each $b \in L$, $P_{\varphi_d(b)}$ has unique solution X_b. □

Next, we prove Kleene's theorem, which states that HYP $= \Delta^1_1$. This is the effective analog of Suslin's theorem, which states that Borel $= \boldsymbol{\Delta^1_1}$.

Theorem 3.6. HYP $\subseteq \Delta^1_1$.

Proof. The point is that both the Σ^1_1 and Π^1_1 sets are uniformly closed under jump, Turing reducibility, and recursive join. This allows us to prove that HYP $\subseteq \Delta^1_1$ by effective transfinite recursion. For details, see [3, §5.2]. □

For the reverse inclusion, we follow Moschovakis's [45, Theorem 3E.1] presentation of Spector's proof of Kleene's theorem.

Definition 3.7. For each e, let L_e denote the e^{th} computable linear ordering. Let $W \subseteq \mathbb{N}$ be the set of all indices for computable well-orderings. For each $a \in W$, let $W_a \subseteq W$ be the set of all indices for computable well-orderings which embed into a proper initial segment of L_a.

Clearly W is Π^1_1. As for W_a:

Proposition 3.8. *For each $a \in W$, W_a is Δ^1_1.*

Proof. TFAE:

- $b \in W_a$;
- there is some embedding from L_b into a proper initial segment of L_a;
- $b \in W$ and there is no embedding from L_a into L_b.

The second clause is Σ^1_1 and the third clause is Π^1_1. □

This suggests that one can "enumerate" W in ω_1^{CK} many steps, such that at each step of the enumeration, one has only enumerated a Δ_1^1 set. Hence the analogy

$$\begin{array}{lcl} \Pi_1^1 & \sim & \text{recursively enumerable} \\ \Delta_1^1 & \sim & \text{finite.} \end{array}$$

This analogy is explored further in metarecursion theory, see Sacks [51, Chapter V].

The following result is a useful step towards proving that $\Delta_1^1 \subseteq \mathrm{HYP}$.

Theorem 3.9. *For each $a \in W$, W_a is hyperarithmetic.*

We give an (undoubtedly mistake-riddled) sketch.

Sketch. Fix a computable well-ordering L. Let $\langle X_a\rangle_{a\in L}$ be the jump hierarchy along L. We use L'-effective transfinite recursion along L to define a total L'-recursive function $f : L \to \mathbb{N}$ such that for each $b \in L$, $\Phi_{f(b)}^{X_b}$ is total and defines $\{e : L_e \text{ properly embeds into } L \upharpoonright b\}$.

For $b = 0_L$, do the obvious.

For b which is a successor in L, observe that L_e properly embeds into $L \upharpoonright b$ if and only if for all $c \in L_e$, there is some $a <_L b$ such that $L_e \upharpoonright c$ properly embeds into $L \upharpoonright a$.

For b which is a limit in L, observe that L_e properly embeds into $L \upharpoonright b$ if and only if there is some $a <_L b$ such that L_e properly embeds into $L \upharpoonright a$. □

In order to extend the above result to all Δ_1^1 sets, the following result is useful:

Theorem 3.10. *W is Π_1^1-complete, i.e., W is Π_1^1 and every Π_1^1 set is many-one reducible to W.*

Proof. First we need a normal form for Π_1^1 predicates. Let A be a Π_1^1 subset of $\mathbb{N}$. Then one can show that there is some computable predicate R such that

$$x \in A \quad \Leftrightarrow \quad (\forall f \in \mathbb{N}^{\mathbb{N}})(\exists n)R(f \upharpoonright n, x),$$

and if $R(\sigma, x)$ holds and σ is an initial segment of τ, then $R(\tau, x)$ holds as well.

Next, for each $x \in \mathbb{N}$, consider the computable tree

$$T_x = \{\sigma \in \mathbb{N}^{<\mathbb{N}} : \neg R(\sigma, x)\}.$$

Observe that $x \in A$ if and only if T_x is well-founded.

The connection from trees to linear orderings is given by the Kleene-Brouwer ordering (also known as the Luzin-Sierpinski ordering). If T is a tree, we define a linear ordering $<_{\mathrm{KB}(T)}$ as follows. For any $\sigma, \tau \in T$, we say that $\sigma <_{\mathrm{KB}(T)} \tau$ if one of the following hold:

– τ is an initial segment of σ;
– σ and τ are incomparable in T and σ is to the left of τ, i.e., if n is the least number such that $\sigma(n) \neq \tau(n)$, then $\sigma(n) < \tau(n)$.

One can check that T is well-founded if and only if $\mathrm{KB}(T)$ is well-ordered. This yields a many-one reduction from A to W: $x \in A$ if and only if $\mathrm{KB}(T_x) \in W$. □

Theorem 3.11 (Spector's Σ^1_1-boundedness)**.** *If $B \subseteq \mathbb{N}$ is Σ^1_1 and is a subset of W, then B is contained in W_a for some $a \in W$.*

Proof. Let P_e be the e^{th} Π^1_1 subset of $\mathbb{N}$. Then $\{e : e \in P_e\}$ is Π^1_1, so we can fix a many-one reduction g from it to W, i.e.,

$$e \in P_e \quad \Leftrightarrow \quad g(e) \in W.$$

Given a Σ^1_1 set $B \subseteq W$, consider the Σ^1_1 set

$$S = g^{-1}(\{a : L_a \text{ embeds into } L_b \text{ for some } b \in B\}).$$

If B is the complement of P_i, then we can compute an index $f(i)$ such that S is the complement of $P_{f(i)}$. We prove that $B \subseteq W_{g(f(i))}$.

Since $B \subseteq W$, it follows that $g(S) \subseteq W$. By choice of g, for any $e \in S$, $e \in P_e$. Since S is the complement of $P_{f(i)}$, it follows that $f(i) \in P_{f(i)}$. (If $f(i) \notin P_{f(i)}$, then $f(i) \in S$, but then $f(i) \in P_{f(i)}$ after all.) This implies two facts:

(1) $f(i) \notin S$, i.e., $L_{g(f(i))}$ does not embed into L_b for any $b \in B$.
(2) $g(f(i)) \in W$.

We conclude that $B \subseteq W_{g(f(i))}$. □

Theorem 3.12. $\Delta^1_1 \subseteq \mathrm{HYP}$.

Proof. Let $A \subseteq \mathbb{N}$ be Δ^1_1. Since A is Π^1_1 and W is Π^1_1-complete, there is some many-one reduction g such that $e \in A$ if and only if $g(e) \in W$.

Next, since A is Σ^1_1, the set $g(A)$ is Σ^1_1 as well. Since $g(A) \subseteq W$, by Σ^1_1-boundedness, there is some $\alpha < \omega_1^{CK}$ such that $g(A) \subseteq W_\alpha$.

We conclude that $e \in A$ if and only if $g(e) \in W_\alpha$. Since W_α is hyperarithmetic, this implies that A is hyperarithmetic as well. □

Definition 3.13. Define the problem of *unique closed choice* in a computable metric space X, written UC_X, as follows: given a singleton in X (represented negatively as a closed set), produce the unique element in the singleton.

Example 3.14. $\mathsf{UC}_{\mathbb{N}} \equiv_{sW} \mathsf{C}_{\mathbb{N}}$ and $\mathsf{UC}_{2^{\mathbb{N}}}$ is computable.

Proposition 3.15. $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ *is strongly Weihrauch equivalent to the following problem: given an ill-founded subtree of $\mathbb{N}^{<\mathbb{N}}$, produce any path on the tree.*

$\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$ *is strongly Weihrauch equivalent to the following problem: given an ill-founded subtree of* $\mathbb{N}^{<\mathbb{N}}$ *with a unique path, produce said path.*

Proposition 3.16. *If* $T \subseteq \mathbb{N}^{<\mathbb{N}}$ *has a unique path, then said path is* T*-hyperarithmetic.*

Proof. If X is the unique path on T, then $x \in X$ if and only if there exists some path P on T such that $x \in P$ if and only if for every path P on T, $x \in P$. Hence X is Δ^1_1 in T. We conclude that X is T-hyperarithmetic. □

Remark 3.17. One can leverage the above fact to show that if $T \subseteq \mathbb{N}^{<\mathbb{N}}$ has no T-hyperarithmetic path, then T must contain a perfect tree. Hence T must have continuum many paths. See, for example, [51, III.6.2].

Proposition 3.18. *There is some computable ill-founded tree* $T \subseteq \mathbb{N}^{<\mathbb{N}}$ *which has no hyperarithmetic path.*

Proof. Observe that the predicate $X \in \mathrm{HYP}$ is Π^1_1. ($X \in \mathrm{HYP}$ if and only if there exists some $e \in W$ such that every jump hierarchy on L_e computes X.) Therefore the predicate $X \notin \mathrm{HYP}$ is Σ^1_1. We can put this predicate in the normal form

$$(\exists f \in \mathbb{N}^\mathbb{N})(\forall n) R(f \upharpoonright n, X \upharpoonright n)$$

for some recursive predicate R, with the property that if $R(\sigma_0, \sigma_1)$ holds and $\sigma_0 \preceq \tau_0$ and $\sigma_1 \preceq \tau_1$, then $R(\tau_0, \tau_1)$ holds as well. Consider then the tree T consisting of all $\langle \sigma_0, \sigma_1 \rangle$ such that $R(\sigma_0, \sigma_1)$ fails to hold. T is computable, ill-founded, and every path on T computes some $X \notin \mathrm{HYP}$ (project to the second component). □

Corollary 3.19. $\mathsf{C}_{\mathbb{N}^\mathbb{N}} \not\leq_c \mathsf{UC}_{\mathbb{N}^\mathbb{N}}$.

4. Higher levels of the Weihrauch lattice

Thus far, we have studied several problems which correspond to theorems at the level of ACA_0 or below, such as IVT, WKL, KL, RT^n_k, BWT. Roughly speaking, we have the following correspondence between problems and theorems in reverse mathematics:

- RCA_0 corresponds to the computable problems;
- WKL_0 corresponds to WKL;
- ACA_0 corresponds to lim and finite compositions of lim.

We have seen exceptions to the above correspondences (e.g., IVT), but these are the exception rather than the norm.

How about problems corresponding to theorems which are strictly stronger than ACA_0? The next highest step of the Big Five is ATR_0, so that is a natural place to start. The study of the Weihrauch lattice at this higher level was initiated by Marcone in 2015. Examples of statements at the level of ATR_0 are:

- comparability of well-orderings;
- Ulm's theorem on invariants of abelian p-groups;
- the perfect tree theorem;
- Lusin's separation of analytic sets;
- open determinacy;
- the open Ramsey theorem;
- the König duality theorem on matchings and covers of infinite bipartite graphs.

Examples of statements slightly below ATR_0 (but still stronger than ACA_0) are:

- Σ^1_1-choice;
- Δ^1_1-comprehension.

Let us formulate some problems which correspond to the above statements. First, we formulate a problem which corresponds to ATR_0 itself.

Definition 4.1. Define ATR to be the following single-valued problem: given a pair (L, A) where L is a well-ordering and $A \subseteq \mathbb{N}$, produce the jump hierarchy $\langle X_a \rangle_{a \in L}$ which starts with A.

There are significant differences between the problem ATR and the system ATR_0 in reverse mathematics, as expounded in the remark after Theorem 3.2 in Kihara, Marcone, Pauly [42]. For example, in the setting of reverse mathematics, different models may disagree on which linear orderings are well-orderings.

Theorem 4.2. $\mathsf{ATR} \leq_W \mathsf{UC}_{\mathbb{N}^\mathbb{N}}$.

Proof. By Theorem 3.4, given some computable well-ordering L and some $A \subseteq \mathbb{N}$, we can uniformly compute an index e for the jump hierarchy on L which starts with A as a $\Pi^{0,L\oplus A}_2$-singleton. That means that the jump hierarchy on L which starts with A is the unique X which satisfies the $\Pi^{0,L\oplus A}_2$-formula $\forall x \exists y R^{L\oplus A}_e(x, y, X)$, where R_e denotes the e^{th} computable predicate.

Given e, we can produce an index for a $\Pi^{0,L\oplus A}_1$-singleton by "Skolemizing" as follows. We say that $f : \mathbb{N} \to \mathbb{N}^{<\mathbb{N}}$ is the *minimal Skolem function* which witnesses that X satisfies $\forall x \exists y R^{L\oplus A}_e(x, y, X)$ if for each x,

- $R^{L\oplus A}_e(x, f(x)(0), X)$ holds;
- $|f(x)| = f(x)(0)$;

– for each $w < f(x)(0)$, $f(x)(w)$ is the least number such that $R_e^{L\oplus A}(x, w, X \upharpoonright f(x)(w))\downarrow$ and fails to hold.

Then (f, X) is the unique solution to the $\Pi_1^{0,L\oplus A}$ predicate "f is the minimal Skolem function witnessing that X satisfies $\forall x \exists y R_e^{L\oplus A}(x, y, X)$". This allows us to uniformly compute a subtree T of $\mathbb{N}^{<\mathbb{N}}$ with a unique path, such that the projection to the second component of the path is the desired jump hierarchy. □

Next, we formulate a problem which corresponds to comparability of well-orderings:

Definition 4.3. Define CWO to be the following single-valued problem: given a pair (L, M) of well-orderings, produce either an embedding of L onto an initial segment of M, or an embedding of M onto a proper initial segment of L.

Friedman (see [54, notes for Theorem V.6.8, pg. 199]) showed that comparability of well-orderings is equivalent to ATR_0.

Proposition 4.4. $\mathsf{CWO} \leq_W \mathsf{ATR}$.

Proof. Given (L, M), define N by adding a first element 0_N and a last element m_N to L. We can use ATR to obtain a hierarchy $\langle X_a \rangle_{a\in N}$ such that:

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

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

To define f, if we are given any $b \in L$ and $f \upharpoonright \{a : a <_L b\}$, we need to define $f(b)$, specifically $\Phi_{f(b)}^{X_b}(0)$. Use $X_b = (\bigoplus_{a<_L b} X_a)'''$ to compute whether $\{\Phi_{f(a)}^{X_a}(0) : a <_L b, \Phi_{f(a)}^{X_a}(0)\downarrow\}$ is a proper subset of M. (This is where $X_{0_N} = L \oplus M$ comes in useful, because X_a uniformly computes X_{0_N} for each $a \in L$.) If so, we compute and output the $<_M$-least element of their difference; otherwise diverge. This completes the definition of $\Phi_{f(b)}^{X_b}$.

Apply the recursion theorem to the definition above to obtain a partial recursive function $f : L \to \mathbb{N}$. By transfinite induction along L, for each $b \in L$, if $\Phi_{f(b)}^{X_b}(0)\downarrow$, then:

– $\Phi_{f(a)}^{X_a}(0)\downarrow$ for all $a <_L b$;

– $\Phi^{X_b}_{f(b)}(0)$ is the $<_M$-least element of
$M \backslash \{\Phi^{X_a}_{f(a)}(0) : a <_L b, \Phi^{X_a}_{f(a)}(0)\downarrow\}$,

while if $\Phi^{X_b}_{f(b)}(0)\uparrow$, then $M \backslash \{\Phi^{X_a}_{f(a)}(0) : a <_L b, \Phi^{X_a}_{f(a)}(0)\downarrow\}$ is empty.

To complete the definition of the backward reduction, we consider the following cases.

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

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

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

Next, we work towards showing that $\mathsf{ATR} \leq_W \mathsf{CWO}$.

Definition 4.5. Let Q be the following problem: given well-orderings L and M, decide whether $L < M$ or $M \leq L$.

Theorem 4.6. $\mathsf{UC}_{\mathbb{N}^\mathbb{N}} \leq_W \widehat{\mathsf{Q}}$.

Proof. Suppose we are given some $T \subseteq \mathbb{N}^{<\mathbb{N}}$ which has a unique path. For each $\sigma \in T$, consider the trees

$$S_\sigma = \{\tau \in T : \tau \text{ does not extend } \sigma\}$$
$$T_\sigma = \{\tau \in T : \tau \text{ and } \sigma \text{ are comparable}\}.$$

If σ lies on the unique path on T, then S_σ is well-founded and T_σ is ill-founded. Otherwise, S_σ is ill-founded and T_σ is well-founded. So we could compute the unique path on T if we were able to compare $\mathrm{KB}(S_\sigma)$ and $\mathrm{KB}(T_\sigma)$, for each $\sigma \in T$. However, Q can only compare well-orderings.

In order to overcome this issue, we consider the *double descent tree* of a pair of linear orderings. Given any linear orderings L and M, define $L * M$ to be the Kleene-Brouwer ordering of the tree of finite sequences of the form $\langle (a_0, b_0), \dots, (a_k, b_k) \rangle$ such that $a_0 >_L \cdots >_L a_k$ and $b_0 >_M \cdots >_M b_k$. Then the following hold:

- If either L or M is a well-ordering, then so is $L * M$.
- If L is a well-ordering but M is not, then L embeds into $L * M$.
- If M is a well-ordering, then $L * M$ embeds into $\eta * M$, where η denotes a computable copy of the rational numbers.

Proofs of the first two facts can be found in Simpson [54, Lemma V.6.5]. The third fact (and its usage in this proof) is due to Kihara, Marcone, Pauly [42, Lemma 2.7][4].

Next, consider the following well-orderings:

$$(\eta * M) * L$$
$$(((\eta * M) * L) + 1) * M.$$

If L is well-ordered but M is not, then

$$(\eta * M) * L < ((\eta * M) * L) + 1 \leq (((\eta * M) * L) + 1) * M.$$

If M is well-ordered but L is not, then

$$((\eta * M) * L) * M \leq \eta * M \leq (\eta * M) * L.$$

Therefore, for each $\sigma \in T$, if we define $L = \mathrm{KB}(S_\sigma)$ and $M = \mathrm{KB}(T_\sigma)$, we may apply Q to compare the above pair of well-orderings. This allows us to compute the unique path on T. $\square$

Remark 4.7. Another proof of the above result can be derived from Greenberg, Montalbán [33, Proposition 2.6].

Proposition 4.8. $\widehat{\mathsf{Q}} \leq_W \mathsf{CWO}$.

Proof. Suppose we are given a $\widehat{\mathsf{Q}}$-instance $(L_n, M_n)_n$. Let $N = 1 + \sum_n (L_n + M_n)$. Apply CWO to the following well-orders:

$$\sum_n (L_n + N \cdot \omega)$$
$$\sum_n (M_n + N \cdot \omega).$$

Note that for each n, $L_n + N \cdot \omega$ and $M_n + N \cdot \omega$ are both isomorphic to $N \cdot \omega$. Hence the above well-orders are isomorphic. Furthermore, given an isomorphism from $\sum_n (L_n + N \cdot \omega)$ to $\sum_n (M + N \cdot \omega)$, we can restrict it to obtain isomorphisms from each $L_n + N \cdot \omega$ to $M_n + N \cdot \omega$. That allows us to compute whether $L_n < M_n$: $L_n < M_n$ if and only if the first element of the first copy of N is mapped into M_n by the isomorphism. $\square$

Corollary 4.9 (Kihara, Marcone, Pauly). $\mathsf{ATR} \equiv_W \mathsf{UC}_{\mathbb{N}^\mathbb{N}} \equiv_W \widehat{\mathsf{Q}} \equiv_W \mathsf{CWO}$.

Corollary 4.10. ATR, $\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$, *and* CWO *are parallelizable.*

Next, we turn our attention to weak comparability of well-orderings:

[4] I took the liberty of modifying their proof slightly. Any mistakes are my own.

Definition 4.11. Define WCWO to be the following problem: given a pair (L, M) of well-orderings, produce either an embedding from L into M or an embedding from M into L.

Friedman and Hirst [30] showed that in reverse mathematics, weak comparability of well-orderings is equivalent to ATR_0.

Question 4.12 (Marcone)**.** Do we have $\mathsf{ATR} \equiv_W \mathsf{UC}_{\mathbb{N}^\mathbb{N}} \equiv_W \mathsf{WCWO}$?

Observe that:

Proposition 4.13. $\mathsf{Q} \leq_W \mathsf{WCWO}$.

Proof. Given (L, M), apply WCWO to $(L \cdot \omega + 1, M \cdot \omega)$. If $L \cdot \omega + 1 \leq M \cdot \omega$, then $L < M$. Otherwise, $M \cdot \omega \leq L \cdot \omega + 1$, which implies that $M \leq L$. ☐

It follows from Theorems 4.2, 4.6, and Proposition 4.4 that

Theorem 4.14 (Kihara, Marcone, Pauly)**.** $\mathsf{UC}_{\mathbb{N}^\mathbb{N}} \equiv_W \widehat{\mathsf{WCWO}}$.

5. $\mathsf{ATR} \leq_W \mathsf{WCWO}$

In this section, we show that $\mathsf{ATR} \equiv_W \mathsf{UC}_{\mathbb{N}^\mathbb{N}} \equiv_W \mathsf{WCWO}$. First, we need to figure out how to extract an infinite amount of useful information from a single embedding between two well-orderings.

As a warm-up:

Proposition 5.1 (essentially Shore [52])**.** *There is a computable well-ordering L of ordertype ω^2 such that from any embedding from ω^2 into L, we can uniformly compute $\emptyset'$.*

Proof. Fix a computable 1-1 enumeration $k : \mathbb{N} \to \mathbb{N}$ of $\emptyset'$. We say that t is a *true stage* if after stage t, every number enumerated by k lies above $k(t)$, i.e., $\emptyset'_t \upharpoonright k(t) = \emptyset' \upharpoonright k(t)$. Observe that for each n, there is an n^{th} true stage. Let the true stage function denote the function which maps n to the n^{th} true stage.

The set of true stages is Π^0_1 and uniformly computes $\emptyset'$. Furthermore, any function which majorizes the true stage function uniformly computes $\emptyset'$. (If h majorizes the true stage function, then for each n, $n \in \emptyset'$ if and only if $n \in \emptyset'_{h(n)}$.)

Now, we construct L as follows. For each t, define L_t to be the set of stages $s \geq t$ at which t appears to be a true stage, ordered by the natural number ordering. Observe that if t is indeed a true stage, then L_t has ordertype ω, otherwise L_t is finite. Define $L = \sum_t L_t$. Since there are infinitely many true stages, L has ordertype ω^2.

Suppose we are given an embedding f from ω^2 into L. For each n, if f sends the first element of the $(n+1)^{\text{st}}$ copy of ω into L_t, define $h(n) = t$. Then one can show by induction that $h : \mathbb{N} \to \mathbb{N}$ majorizes the true stage function, and hence uniformly computes $\emptyset'$. $\square$

There are several ideas that make the above proof work. The first idea is that of computing $\emptyset'$ by majorizing its true stage function. This overcomes the basic problem with a coding strategy: if we put coding locations in the target well-ordering, an embedding could skip above our coding locations. This idea of computing via majorization can be generalized to compute jump hierarchies, as we will see.

The second idea is to exploit certain order-theoretic properties of ω, specifically:

> If $\omega \cdot k$ embeds into a finite sum of well-orderings, some of which have ordertype ω and some of which have ordertype $< \omega$, there must be at least k many orderings in the sum with ordertype ω.

More generally, the above property holds for *indecomposable* well-orderings:

Definition 5.2. A well-ordering M is *indecomposable* if it embeds into every final segment of itself.

Lemma 5.3. *Let L be a linear ordering and let M be an indecomposable well-ordering which does not embed into L. If F embeds M into a finite sum of L's and M's, then the range of M under F must be cofinal in some copy of M.*

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

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

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

Indecomposable well-orderings played an essential role in Friedman and Hirst's [30] proof that WCWO implies ATR_0 in reverse mathematics.

Next, we show how to reduce the problem of computing a jump hierarchy into the problem of comparing an indecomposable well-ordering with a sequence of well-orderings. (We did something similar in order to prove Theorem 4.6, but it is not clear whether that approach can be modified to yield this result.)

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

Definition 5.4. A *labeled well-ordering* is a tuple $\mathcal{L} = (L, 0_L, S, p)$ where L is a well-ordering, 0_L is the first element of L, S is the set of all successor elements in L, and $p : S \to L$ is the predecessor function.

Proposition 5.5 (Goh)**.** ATR *is Weihrauch equivalent to the following problem: instances are pairs $(\mathcal{L}, c)$ where $\mathcal{L}$ is a labeled well-ordering and $c \in L$, with unique solution being Y_c, where $\langle Y_a\rangle_{a\in L}$ is the unique hierarchy such that:*

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

Next, we present a uniform analog of a theorem of Chen [16]. (Chen's results concern the many-one degree of W_e, for each $e \in W$.) Our proof is adapted from Shore [52, Theorem 3.5].

Theorem 5.6 (Goh)**.** *Given a labeled well-ordering $\mathcal{L}$, we can uniformly compute an indecomposable well-ordering M and well-orderings $\langle K(a, n)\rangle_{n\in\mathbb{N},a\in L}$ such that:*

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

In order to prove the above theorem, we define some computable operations on trees.

Definition 5.7 (Shore [52, Definition 3.9], slightly modified)**.** For any (possibly finite) sequence of trees $\langle T_i\rangle$, we define their *maximum* by joining all T_i's at the root, i.e.,

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

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

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

Lemma 5.8 (Shore [52, Lemma 3.10])**.** *Let $\langle T_i \rangle_i$ be a (possibly finite) sequence of trees.*

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

With the maximum and minimum operations in hand, we may prove an analog of Theorem 3.11 in Shore [52]:

Theorem 5.9. *Given a labeled well-ordering $\mathcal{L}$, we can uniformly compute sequences of trees $\langle g(a,n) \rangle_{n \in \mathbb{N}, a \in L}$ and $\langle h(a,n) \rangle_{n \in \mathbb{N}, a \in L}$ such that:*

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

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

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

For $b = a+1$, fix a recursively enumerable set W which enumerates X' from X for any X. In particular,

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

Then define

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

If $n \in Y_b$, then there is some $\langle P, Q, n \rangle \in W$ such that $P \subseteq Y_a$ and $Q \subseteq Y_a^c$. Then every tree in the above minimum for $\langle P, Q, n \rangle$

is ill-founded, so the minimum is itself ill-founded. Hence $h(b,n)$ is ill-founded.

If $n \notin Y_b$, then for all $\langle P, Q, n\rangle \in W$, either $P \not\subseteq Y_a$ or $Q \not\subseteq Y_a^c$. Either way, all of the above minima have rank $< \omega \cdot \mathrm{otp}(L \upharpoonright a) + \omega$. Hence $h(b,n)$ has rank at most $\omega \cdot \mathrm{otp}(L \upharpoonright a) + \omega = \omega \cdot \mathrm{otp}(L \upharpoonright b)$.

Similarly, define

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

This completes the construction for the successor case. □

Next, we adapt the above construction to obtain well-founded trees. To that end, for each well-ordering L, we aim to compute a tree $(T(\omega \cdot L))^\infty$ which is universal for all trees of rank $\leq \omega \cdot \mathrm{otp}(L)$. Shore [52, Definition 3.12] constructs such a tree by effective transfinite recursion. Instead, we use a simpler construction of Greenberg and Montalbán [33].

Definition 5.10. Given a linear ordering L, define $T(L)$ to be the tree of finite $<_L$-decreasing sequences, ordered by extension.

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

Definition 5.11 ([33, Definition 3.20])**.** Given a tree T, define a tree

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

ordered by extension.

Lemma 5.12 (essentially [33, §3.2.2])**.** *Let T be well-founded. Then*

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

Finally, we prove our analog of Chen's theorem.

Proof of Theorem 5.6. Given $\mathcal{L}$, we may use Theorem 5.9, Definition 5.10 and Definition 5.11 to uniformly compute

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

By Lemma 5.12(4), M is indecomposable. We want to show that:

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

– if $n \notin Y_a$, then $K(a, n) < M$.

First,

$$\begin{aligned} &\operatorname{rk}(T(\omega \cdot L)^\infty) = \omega \cdot \operatorname{otp}(L) \\ \text{so} \qquad &\operatorname{rk}(\min\{T(\omega \cdot L)^\infty, h(a,n)\}) \leq \omega \cdot \operatorname{otp}(L). \end{aligned}$$

It then follows from Lemma 5.12(5) that $K(a, n) \leq M$.

If $n \in Y_a$, then $h(a, n)$ is ill-founded. Fix some descending sequence $\langle \sigma_i \rangle_i$ in $h(a, n)$. Then we may embed $T(\omega \cdot L)^\infty$ into $\min\{T(\omega \cdot L)^\infty, h(a, n)\}$ while preserving $<_{\mathrm{KB}}$: map τ to $\langle\langle \tau \restriction i, \sigma_i \rangle\rangle_{i=0}^{|\tau|}$. Therefore $M \leq K(a, n)$, showing that $K(a, n) \equiv M$ in this case.

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

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

Since $\omega \cdot \operatorname{otp}(L \restriction a) + 1 < \omega \cdot \operatorname{otp}(L)$, by Lemma 5.12(5), $K(a, n) < M$. □

The final ingredient, presented below, will allow us to compute Y_a by majorizing an appropriate function, just as we computed $\emptyset'$ by majorizing its true stage function. (For more on this topic, see Slaman and Groszek [55] and Gerdes's thesis [31].)

Proposition 5.13 (essentially Jockusch, McLaughlin [39, Theorem 3.1])**.** *Given a labeled well-ordering $\mathcal{L}$ and $a \in L$, we can uniformly compute an index for a $\Pi^{0,\mathcal{L}}_1$-singleton $\{f\}$ which is strictly increasing, and Turing reductions witnessing that $f \equiv_T Y_a$.*

Proof. This result can be obtained by analyzing the reduction $\mathsf{ATR} \leq_W \mathsf{UC}_{\mathbb{N}^\mathbb{N}}$. Given $\mathcal{L}$ and $a \in L$, we can uniformly compute an index for Y_a as a $\Pi^{0,\mathcal{L}}_2$-singleton. Then we define f to be the join of Y_a and the lex-minimal Skolem function F which witnesses that Y_a satisfies the $\Pi^{0,\mathcal{L}}_2$ predicate that we computed. We can uniformly compute an index for f as a $\Pi^{0,\mathcal{L}}_1$-singleton. Also, f computes Y_a by projection.

It remains to compute an index for a Turing reduction from F to Y_a. The point is that $\mathcal{L} \oplus Y_a$ can compute F by exhaustive search. Also, Y_a uniformly computes $Y_{0_\mathcal{L}} = \mathcal{L}$. We conclude that Y_a uniformly computes $f = Y_a \oplus F$.

Finally, we replace $f : \mathbb{N} \to \mathbb{N}$ with the strictly increasing function $n \mapsto \sum_{m \leq n}(f(m) + 1)$. For the new f, we can uniformly compute indices for it as a $\Pi^{0,\mathcal{L}}_1$-singleton, and Turing reductions witnessing that $f \equiv_T Y_a$. □

Proposition 5.14 (see [39, Lemma 4.9(2)])**.** *If $\{f\}$ is a $\Pi^{0,\mathcal{L}}_1$-singleton and g majorizes f, then $\mathcal{L} \oplus g$ uniformly computes f.*

Proof. This follows from König's lemma. Think of f as the unique path on an $\mathcal{L}$-computable tree T. If g majorizes f, then the g-bounded subtree of T is a finitely branching $\mathcal{L}$-computable tree with a unique path f. From $\mathcal{L} \oplus g$, we can compute f inductively, by waiting for all other g-bounded strings in T to die out. □

Finally, we combine Theorem 5.6 with the above results to prove that

Theorem 5.15 (Goh)**.** $\mathsf{ATR} \leq_W \mathsf{WCWO}$.

Proof. We reduce the version of ATR in Proposition 5.5 to WCWO. Given a labeled well-ordering $\mathcal{L}$ and $a \in L$, there is some strictly increasing f such that if g majorizes f, then $\mathcal{L} \oplus g$ uniformly computes Y_a.

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

Next, use $\mathcal{L}$ to compute labels for $(L \restriction \{b : b \leq_L a\}) + 1$. Apply Theorem 5.6 to $(L \restriction \{b : b \leq_L a\}) + 1$ (and its labels) to compute an indecomposable well-ordering M and for each n, a well-ordering $L_n := K(a+1, r(n))$, such that

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

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

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

Since M is indecomposable, $L_n \leq M$ for all n, and there are infinitely many n such that $L_n \equiv M$, it follows that $\sum_n L_n$ has the same ordertype as $\sum_n M$. Hence any WCWO-solution F must go from left to right. Furthermore, since M is indecomposable, it has no last element, so F must embed $\sum_n M$ into $\sum_n L_n$.

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

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

We show that g majorizes f. For each n, F embeds $M \cdot n$ into $\sum_{i \leq g(n)} L_i$. It follows from Lemma 5.3 that at least n of the L_i's for $i \leq g(n)$ must have ordertype M. That means that there must be at least n elements in the range of f which lie below $g(n)$, i.e., $f(n) \leq g(n)$.

Since g majorizes f, $\mathcal{L} \oplus g$ uniformly computes Y_a, as desired. $\square$

It follows from Theorem 5.15 and Proposition 4.4 that

Corollary 5.16 (Goh)**.** $\mathsf{CWO} \equiv_W \mathsf{ATR} \equiv_W \mathsf{WCWO}$.

6. The König duality theorem

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

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

Theorem 6.1. *For any bipartite graph G, there is a matching M and a cover C which are* dual*, i.e., C is obtained by choosing exactly one vertex from each edge in M. Such a pair (C, M) is said to be a* König cover*.*

König proved the above theorem for finite graphs, where it is commonly stated as "the maximum size of a matching is equal to the minimum size of a cover". For infinite graphs, this latter form would have little value. Instead of merely asserting the existence of a bijection, we want such a bijection to respect the structure of the graph. Hence the notion of a König cover. Podewski and Steffens [50] proved König's duality theorem for countable graphs. Finally, Aharoni [1] proved it for graphs of arbitrary cardinality. In this course, we will only study the theorem for countable graphs.

Definition 6.2. KDT is the following problem: given a (countable) bipartite graph G, produce a König cover (C, M).

Note that we represent bipartite graphs as their vertex set and edge relation. Alternatively, our representation of a bipartite graph could also include a partition of its vertex set which witnesses that the graph is bipartite. Even though these two representations are not computably equivalent[5], all of our results hold for either representation.

[5] In fact, there is a computable bipartite graph such that no computable partition of its vertices witnesses that the graph is bipartite. This was known to Bean [4, remarks after Theorem 7] (we thank Jeff Hirst for pointing this out.) See also Hirst [38, Corollary 3.17].

Aharoni, Magidor, Shore [2] studied König's duality theorem for countable graphs from the point of view of reverse mathematics. They showed that ATR_0 is provable from König's duality theorem. They also showed that König's duality theorem is provable in the system Π^1_1-CA_0, which is strictly stronger than ATR_0. Simpson [53] then closed the gap by showing that König's duality theorem is provable in (hence equivalent to) ATR_0.

We now translate the proof of ATR_0 from König's duality theorem in [2] into a Weihrauch reduction from ATR to KDT. For our forward reduction, the bipartite graphs we construct will be sequences of subtrees of $\mathbb{N}^{<\mathbb{N}}$. Let us define our notation regarding trees. For us, a *rooted subtree* of $\mathbb{N}^{<\mathbb{N}}$ is a subset T of $\mathbb{N}^{<\mathbb{N}}$ for which there is a unique $r \in T$ (called the *root*) such that:

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

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

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

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

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

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

Definition 6.3. Given a tree T (with root r) and a König cover (C, M) of T, we can decode the bit b, which is the Boolean value of $r \in C$. We say that (C, M) *codes* b.

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

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

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

Note that every König cover of a disjoint union of graphs restricts to a König cover for each graph in the disjoint union. Therefore we will not distinguish between a König cover of the disjoint union of a sequence of trees, and a sequence of König covers, one for each of the trees in the sequence.

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

Definition 6.4. A tree T is *good* if its root r lies in C for every König cover (C, M) of T, or lies outside C for every König cover (C, M) of T. A sequence of trees $\langle T_n \rangle$ is *good* if every T_n is good. In other words, $\langle T_n \rangle$ is good if all of its König covers code the same set.

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

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

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

Lemma 6.5. *Given any $A \subseteq \mathbb{N}$, we can uniformly compute a sequence of trees $\langle T_n \rangle$ which codes A.*

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

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

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

We may use this as the base case for our construction as well. As for the successor case, we will prove

Lemma 6.6. *Given a sequence of trees $\langle T_i : i \in \mathbb{N}\rangle$ (each with the empty node as root), we can uniformly compute a sequence of trees $\langle S_e : e \in \mathbb{N}\rangle$ (each with the empty node as root) such that if $\langle T_i \rangle$ codes a set A, then $\langle S_e \rangle$ codes A'.*

In order to prove the above lemma, we state a sufficient condition on a König cover of a tree and a node in said tree which ensures that the given König cover, when restricted to the subtree above the given node, remains a König cover. The set of all nodes satisfying this condition form a subtree, as follows:

Definition 6.7. For any tree T (with root r) and any König cover (C, M) of T, define the subtree T^* (with root r):

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

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

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

Lemma 6.8 ([2, Lemma 4.5])**.** *For any T and any König cover (C, M) of T, define T^* as above. Then for any $t \in T^*$, (C, M) restricts to a König cover of the subtree of T (not T^*!) above t.*

Proof. It is clear that C restricts to a cover and M restricts to a matching, in the subtree of T above t. It is also clear that no edge in M in the subtree above t has both endpoints in (the restriction of) C.

It remains to show that each $s \in C$ which extends t is the endpoint of some edge in M in the subtree of T above t.

If s strictly extends t, then the desired fact follows from our assumption that (C, M) is a König cover.

If $s = t$, that means that $t \in C$. Since $t \in T^*$, we have that $(t \upharpoonright (|t|-1), t) \notin M$. Since (C, M) is a König cover, there must be some t' immediately extending t such that $(t, t') \in M$, as desired. $\square$

Using Definition 6.7 and Lemma 6.8, we may easily show that:

Proposition 6.9. *Let (C, M) be a König cover of T. Suppose that $t \in T^*$. Let S denote the subtree of T above t. Then S^* is the subtree of T^* above t, where S^* is calculated using the restriction of (C, M) to S.*

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

Definition 6.10. Given a (possibly finite) sequence of trees $\langle T_i \rangle$, each with the empty node as root, we may *combine* it to form a single tree S, by adjoining two copies of each T_i to a root node r. Formally,

$$S = \{r\} \cup \{r^\frown(i,j)^\frown\sigma : \sigma \in T_i, j < 2\}.$$

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

Lemma 6.11. *Suppose $\langle T_i : i \in X \rangle$ combine to form S. Let r denote the root of S, and for each $i \in X$, let $r_{i,0}$ and $r_{i,1}$ denote the roots of the two copies of T_i in S (i.e., $r_{i,0} = r^\frown(i,0)$ and $r_{i,1} = r^\frown(i,1)$). Given any König cover (C, M) of S, for each $i \in X$, we can uniformly computably choose one of $r_{i,0}$ or $r_{i,1}$ (call our choice r_i) such that:*

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

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

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

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

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

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

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

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

Definition 6.12. The *complement* of T, denoted $\overline{T}$, is defined by combining the single-element sequence $\langle T \rangle$.

By Lemma 6.11, if T codes the bit i, then $\overline{T}$ codes the bit $1 - i$.

Lemma 6.13 ([2, Lemma 4.7])**.** *Given a sequence of trees $\langle T_i : i \in \mathbb{N} \rangle$ which codes a set $A \subseteq \mathbb{N}$, we can uniformly compute a sequence of trees $\langle S_e : e \in \mathbb{N} \rangle$ which codes A'.*

Proof. For each e, we construct S_e as follows. Observe that $e \in A'$ if and only if

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

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

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

Theorem 6.14. $\mathsf{ATR} \leq_W \mathsf{KDT}$.

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

For the base case, we use Lemma 6.5 to compute a sequence of trees $\langle T^{0_L}_n\rangle_n$ which codes A. Otherwise, for $b >_L 0_L$, we use Lemma 6.13 to compute a sequence of trees $\langle T^b_n\rangle_n$ such that if for each $a <_L b$, $\Phi^{\mathcal{L}\oplus A}_{f(a)}$ is (interpreted as) a sequence of trees $\langle T^a_n\rangle_n$ which codes Y_a, then $\langle T^b_n\rangle_n$ codes $\left(\bigoplus_{a<_L b} Y_a\right)'$.

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

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

By transfinite induction along L using Lemmas 6.5 and 6.13, $\langle T^b_n\rangle_n$ is good for all $b \in L$, and $\langle Y_b\rangle_{b\in L}$ is the jump hierarchy on L which starts with A. □

Do we have $\mathsf{KDT} \leq_W \mathsf{ATR}$ as well? It turns out this is far from true. In order to prove this, we need to discuss how we represent trees. The usual way to represent a tree (at least in computable structure theory) is by a pair (e, X), where Φ^X_e is total and defines a subset of $\mathbb{N}^{<\mathbb{N}}$ which is a tree.

Instead, we use an alternative representation. For each $r \in \mathbb{N}^{<\mathbb{N}}$, $e \in \mathbb{N}$ and $X \subseteq \mathbb{N}$, (r, e, X) is a name for the following tree T with root node r: $r^\frown\sigma \in T$ if and only if for all $k < |\sigma|$, $\Phi^X_{e,k+\max_{i<k}\sigma(i)}(\sigma \upharpoonright k)\downarrow= 1$.

This representation clearly reduces to the usual representation. The converse may not hold (I don't have a proof that it does not), but they yield equivalent versions of KDT.

Proposition 6.15. *The strong Weihrauch degree of* KDT *for sequences of trees is the same regardless of which of the above two representations we use.*

Proof. It suffices to prove the desired statement for KDT for trees. Suppose we are given some (e, X) such that Φ_e^X is a tree (with empty node as root).

We define another tree T with empty node as root as follows. For each σ such that $\Phi_e^X(\sigma)\downarrow= 1$, define a string σ' as follows: for $k < |\sigma|$, $\sigma'(k)$ is defined to be $\langle \sigma(k), s\rangle$, where s is the least stage such that $\Phi_{e,s}^X(\sigma \upharpoonright (k+1))\downarrow= 1$. It is clear that we can enumerate T sufficiently quickly. For example, for each σ', we can decide by stage $|\sigma'| + \max_{k<|\sigma'|} \pi_1(\sigma'(k))$ whether it should be enumerated into T.

Observe that there is a uniformly computable isomorphism from T to Φ_e^X: map each σ' in T to the string σ, defined by $\sigma(k) = \pi_0(\sigma'(k))$ for each $k < |\sigma'|$. Given a König cover of T, we can uniformly compute a König cover of Φ_e^X via this isomorphism. □

The advantage of our representation is that every (r, e, X) names some tree.

Definition 6.16. A representation $\delta :\subseteq \mathbb{N}^\mathbb{N} \to X$ is *total* if $\mathrm{dom}(\delta) = \mathbb{N}^\mathbb{N}$.

Since we can interpret every $p \in \mathbb{N}^\mathbb{N}$ as some (r, e, X), our representation of trees is total.

Observe that our reduction from ATR to KDT can be modified to work with our alternative representation of trees. (For example, in the proof of Lemma 6.5, in order to code that $n \in A$, we should use the tree $\{\langle\rangle, \langle s\rangle, \langle s+1\rangle\}$, where s is the stage at which n enters A.)

We are ready to prove:

Theorem 6.17 ([2, Theorem 4.12])**.** *There is a computable bipartite graph G such that every König cover of G computes every hyperarithmetic set.*

Proof. For any e which is an index for a computable well-ordering L_e, we have showed how to uniformly construct a sequence of trees which code the jump hierarchy on L_e.

The point is that even if e is an index for a computable ill-founded linear ordering, we can still perform the above construction. Since our representation is total, we obtain a sequence of trees in any case. (We can no longer show by induction that the resulting trees are good, but that does not matter.)

Then, take the disjoint union of the sequences of the trees produced above. Apply KDT to obtain a König cover for each tree. For sequences of trees which are produced from well-orderings L_e, their König covers code the jump hierarchy on L_e. Hence the entire sequence of König covers computes every hyperarithmetic set. $\square$

Corollary 6.18. $\mathsf{KDT} \not\leq_c \mathsf{ATR}$, *hence* $\mathsf{KDT} \not\leq_W \mathsf{ATR}$.

Proof. Every computable instance of ATR has a hyperarithmetic solution, while the above theorem shows that there is a computable instance of KDT with no hyperarithmetic solution. $\square$

7. Interlude: Two-sided problems

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

One obvious way to combine such problems is to take their union. For example, a "two-sided" version of ATR could map a well-ordering to a jump hierarchy on it, and map an ill-founded linear ordering to an infinite descending sequence in it. We will not consider such problems here, because they are not Weihrauch reducible (or even arithmetically Weihrauch reducible) to $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$. (Any such reduction could be used to give a Σ^1_1 definition for the set of indices of pairs of well-orderings. See also Brattka, de Brecht, Pauly [6, Theorem 7.7].) On the other hand, it is not hard to see that all of the problems that we have considered thus far, including KDT, are Weihrauch reducible to $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$.

However, some ill-founded linear orderings support jump hierarchies (known as pseudohierarchies)! This suggests the following two-sided version of ATR.

Definition 7.1. ATR_2 is the following problem: given a linear ordering L and a set $A \subseteq \mathbb{N}$, either produce an infinite $<_L$-descending sequence S, or a jump hierarchy $\langle X_a \rangle_{a \in L}$ on L which begins with A. In either case we indicate which type of solution we produce.

Observe that ATR_2 is Weihrauch reducible to $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$, because it is defined by an arithmetical predicate.

We defer the study of other basic properties of ATR_2 to a later section.

8. Reducing ATR_2 to KDT

Our forward reduction from ATR_2 to KDT will be the same as that from ATR to KDT. By "effective transfinite recursion" along a given linear ordering $\mathcal{L}$, we may construct trees $\langle T^b_n\rangle_{b\in L,n\in\mathbb{N}}$ as before.

If $\mathcal{L}$ is ill-founded, there may be some $a \in L$ and $i \in \mathbb{N}$ such that T^a_i is not good, i.e., there may be some $r, s \in \mathbb{N}^{<\mathbb{N}}$ and some König covers of $r^\frown T^a_i$ and $s^\frown T^a_i$ which code different bits. In order to salvage the situation, we will check for such inconsistencies in the backward reduction. If they are present, we use them to compute an infinite $<_L$-descending sequence.

Before doing so, we need to state a more general and more informative version of [2, Lemma 4.7]. The construction is the same as that in the proof of Lemma 6.13.

Lemma 8.1. *Given a sequence of trees $\langle T_i : i \in \mathbb{N}\rangle$ (each with the empty node as root), we can uniformly compute a sequence of trees $\langle S_e : e \in \mathbb{N}\rangle$ (each with the empty node as root) such that given a König cover (C_e, M_e) of S_e, we can uniformly compute a sequence of sets of nodes $\langle R_{e,i}\rangle_i$ in S^*_e such that*

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

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

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

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

There are several things to point out regarding the statement of the above lemma:

1. For each e and i, instead of choosing a single node r_i as in Lemma 6.11, we now have to choose a set of nodes $R_{e,i}$. This is because we might want to copy the tree T_i more than twice, at multiple levels of the tree S_e. If T_i is not good, these copies could code different bits (according to appropriate restrictions of (C_e, M_e)), so we could have $R_{e,i} \not\subseteq C_e$ and $R_{e,i} \not\subseteq \overline{C_e}$. In that

case, we have little control over whether the root of S_e lies in C_e.

(2) Conclusion (1) will not be needed for our subsequent proofs. It is easily observed from the proof of Lemma 6.13.

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

Suppose that we are given a König cover (C_n^b, M_n^b) of T_n^b. Then we can apply the above lemma to compute, for each $a <_L b$ and $i \in \mathbb{N}$, a set of nodes $R_{n,i}^a$ in $(T_n^b)^*$ such that:

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

$$Y_a = \{i \in \mathbb{N} : R_{n,i}^a \subseteq C_n^b\}.$$

Next, we define the sets $R_{n,i}^{b,a}$ as follows:

Definition 8.2. Fix a labeled linear ordering $\mathcal{L}$ and use the forward reduction in Theorem 6.14 to compute $\langle\langle T_n^b\rangle_n\rangle_{b\in L}$. For each n and b, fix a König cover (C_n^b, M_n^b) of T_n^b. For each $a <_L b$ and each $i, n \in \mathbb{N}$, we define a set of nodes $R_{n,i}^{b,a}$ in T_n^b as follows: $R_{n,i}^{b,a}$ is the set of all r for which there exist $j \geq 1$ and

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

such that for all $0 < l \leq j$, r_l lies in $R_{i_{l-1},i_l}^{c_l}$ as calculated by applying Lemma 6.13 to (C_n^b, M_n^b) restricted to the subtree of T_n^b above r_{l-1}.

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

(1) By induction on l, r_l lies in $(T_n^b)^*$ and the subtree of T_n^b above r_l is $r_l{}^\frown T_{i_l}^{c_l}$. In particular, for each $r \in R_{n,i}^{b,a}$, $r \in (T_n^b)^*$ and the subtree of T_n^b above r is $r^\frown T_i^a$.

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

Definition 8.3. In the same context as the previous definition, we say that $a \in L$ is *consistent* if for all $i \in \mathbb{N}$:

$$\begin{aligned} &\text{the root of } T_i^a \in C_i^a &&\Rightarrow && R_{n,i}^{b,a} \subseteq C_n^b \text{ for all } b >_L a, n \in \mathbb{N} \\ &\text{the root of } T_i^a \notin C_i^a &&\Rightarrow && R_{n,i}^{b,a} \subseteq \overline{C_n^b} \text{ for all } b >_L a, n \in \mathbb{N}. \end{aligned}$$

Observe that if T_i^a is good for all i, then observation (1) above implies that a is consistent, regardless of what $\langle (C_n^b, M_n^b) \rangle_{b,n}$ may be. However, unless L is well-founded, we cannot be certain that T_i^a is good. Consistency is a weaker condition which suffices to ensure that we can still obtain a jump hierarchy on L, as we show in Corollary 8.6. We will also show that inconsistency cannot come from nowhere, i.e., if b_0 is inconsistent, then there is some $b_1 <_L b_0$ which is inconsistent, and so on, yielding an infinite $<_L$-descending sequence of inconsistent elements.

Furthermore, consistency is easy to check: by observation (2) above, whether a is consistent is Π_1^0 (in $\mathcal{L} \oplus \langle (C_n^b, M_n^b) \rangle_{b,n}$).

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

Lemma 8.4. *Fix König covers $\langle (C_n^b, M_n^b) \rangle_{b,n}$ for $\langle T_n^b \rangle_{b,n}$. Now fix n and b. Suppose that for each $a <_L b$, the set $Y_a \subseteq \mathbb{N}$ is such that*

$$\begin{aligned} i \in Y_a &\quad\Rightarrow\quad R_{n,i}^{b,a} \subseteq C_n^b \\ i \notin Y_a &\quad\Rightarrow\quad R_{n,i}^{b,a} \subseteq \overline{C_n^b}. \end{aligned}$$

Then for each n, $n \in \left(\bigoplus_{a <_L b} Y_a\right)'$ if and only if the root of T_n^b lies in C_n^b. In other words, (C_n^b, M_n^b) codes the n^{th} bit of $\left(\bigoplus_{a <_L b} Y_a\right)'$.

Proof. Recall that $\langle T_n^b \rangle_{n \in \mathbb{N}}$ is computed by applying Lemma 6.13 to $\langle \langle T_n^a \rangle_{n \in \mathbb{N}} \rangle_{a <_L b}$. By definition of $R_{n,i}^{b,a}$, $R_{n,i}^a$ (as obtained from Lemma 6.13) is a subset of $R_{n,i}^{b,a}$ (this is the case $j = 1$). So for all $a <_L b$,

$$\begin{aligned} i \in Y_a &\quad\Rightarrow\quad R_{n,i}^a \subseteq R_{n,i}^{b,a} \subseteq C_n^b \\ i \notin Y_a &\quad\Rightarrow\quad R_{n,i}^a \subseteq R_{n,i}^{b,a} \subseteq \overline{C_n^b}. \end{aligned}$$

The desired result follows from Lemma 6.13(3). □

Lemma 8.5. *Fix König covers $\langle (C_m^c, M_m^c) \rangle_{c,m}$ for $\langle T_m^c \rangle_{c,m}$. Now fix m and $b <_L c$. Suppose that for each $a <_L b$, the set $Y_a \subseteq \mathbb{N}$ is such that*

$$\begin{aligned} i \in Y_a &\quad\Rightarrow\quad R_{m,i}^{c,a} \subseteq C_m^c \\ i \notin Y_a &\quad\Rightarrow\quad R_{m,i}^{c,a} \subseteq \overline{C_m^c}. \end{aligned}$$

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

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

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

Proof. If $R^{c,b}_{m,n}$ is empty, then the desired result is vacuously true. Otherwise, consider $r \in R^{c,b}_{m,n}$. As we observed right after Definition 8.2, $r \in (T^c_m)^*$ and the subtree of T^c_m above r is $r^\frown T^b_n$. T^b_n was constructed by applying Lemma 6.13 to $\langle\langle T^a_n\rangle_{n\in\mathbb{N}}\rangle_{a<_L b}$, so we can use the restriction of (C^c_m, M^c_m) to $r^\frown T^b_n$ to compute sets $\langle R^a_{n,i}\rangle_{a<_L b, i\in\mathbb{N}}$ of nodes in $(r^\frown T^b_n)^*$ satisfying the conclusions of Lemma 6.13.

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

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

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

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

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

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

By our claim, we have that

$$i \in Y_a \quad \Rightarrow \quad R^a_{n,i} \subseteq R^{c,a}_{m,i} \subseteq C^c_m$$

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

By Lemma 6.13(3), $n \in \left(\bigoplus_{a<_L b} Y_a\right)'$ if and only if $r \in C^c_m$. This concludes the proof. □

Putting the previous two lemmas together, we obtain

Corollary 8.6. *Fix König covers* $\langle (C^b_n, M^b_n)\rangle_{b,n}$ *for* $\langle T^b_n\rangle_{b,n}$. *For each* $b \in L$, *define* Y_b *by decoding* $\langle (C^b_n, M^b_n)\rangle_n$, *i.e.,*

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

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

Proof. 0_L is consistent because every $T_n^{0_L}$ is good (as constructed in the proof of Lemma 6.5). Consider now any $b >_L 0_L$. Every $a <_L b$ is consistent, so for all $a <_L b$:

$$\begin{aligned} i \in Y_a \quad &\Rightarrow \quad R^{c,a}_{m,i} \subseteq C^c_m \text{ for all } c >_L a, m \in \mathbb{N} \\ i \notin Y_a \quad &\Rightarrow \quad R^{c,a}_{m,i} \subseteq \overline{C^c_m} \text{ for all } c >_L a, m \in \mathbb{N}. \end{aligned}$$

By Lemma 8.4, $Y_b = \big(\bigoplus_{a<_L b} Y_a\big)'$.

Also, by Lemma 8.5, for all $n \in \mathbb{N}$:

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

It follows that b is consistent. □

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

Theorem 8.7 (Goh)**.** $\mathsf{ATR}_2 \leq_W \mathsf{LPO} * \mathsf{KDT}$. *In particular,* $\mathsf{ATR}_2 \leq_c \mathsf{KDT}$ *and* $\mathsf{ATR}_2 \leq_W^{\text{arith}} \mathsf{KDT}$*.*

Proof. Given a labeled linear ordering $\mathcal{L}$ and a set A, we apply the forward reduction in Theorem 6.14 to produce some KDT-instance $\langle T^b_n \rangle_{b,n}$. For the backward reduction, given a KDT-solution $\langle \langle (C^b_n, M^b_n) \rangle_n \rangle_{b \in L}$, we start by uniformly decoding it to obtain a sequence of sets $\langle Y_b \rangle_{b \in L}$.

Next, since $R^{b,a}_{n,i}$ is uniformly c.e. in $\mathcal{L} \oplus (C^b_n, M^b_n)$, whether some $a \in L$ is inconsistent is uniformly c.e. in $\mathcal{L} \oplus \langle (C^b_n, M^b_n) \rangle_{b,n}$. Therefore we can use LPO to determine whether every $a \in L$ is consistent.

If so, by Corollary 8.6, $\langle Y_b \rangle_{b \in L}$ is a jump hierarchy on L which starts with A.

If not, by Corollary 8.6, every inconsistent element is preceded by some other inconsistent element. Since whether some $a \in L$ is inconsistent is uniformly c.e. in $\mathcal{L} \oplus \langle (C^b_n, M^b_n) \rangle_{b,n}$, we can use it to compute an infinite $<_L$-descending sequence of inconsistent elements. □

9. A proof of KDT

In this section, we present a proof of KDT, following Aharoni, Magidor, Shore [2]. Fix a countable bipartite graph G, with sides X and Y.

First of all, it will be helpful to think of our matchings as going in certain directions: we say that F is a *matching from A into B* if every vertex in A is matched to some vertex in B. Note that F is an injection from A into B, so for each x, we will denote the vertex which is matched to x by $F(x)$. The overall strategy consists of two steps:

(1) construct some $A^* \subseteq X$ and a matching from A^* into $D \subseteq Y$;
(2) construct some matching from $Y - D$ into $X - A^*$.

Clearly the union of the above two matchings is itself a matching. As for the cover, we take the union of A^* and $Y - D$. In order for $A^* \cup (Y - D)$ to cover G, we will choose D to be the set of all vertices whose neighbors all lie in A^*. We set up notation for that: for each vertex x, we denote its set of neighbors by $N_G(x)$. For each set A of vertices in X, we define the *demand of* A, denoted $D_G(A)$, to be the set of all vertices whose neighbors all lie in A, i.e.,

$$D_G(A) = \{y \in Y : N_G(y) \subseteq A\}.$$

The set D in the strategy above is in fact $D_G(A^*)$.

Just to get ourselves thinking about the definition, here are some properties:

- if $A \subseteq B \subseteq X$, then $D_G(A) \subseteq D_G(B)$;
- if $\{x, y\} \in E$ and $A \subseteq X$ does not contain x, then $D_{G-\{x,y\}}(A) \supseteq D_G(A)$.

One good reason to consider the demand set is to allow us to build matchings step by step. As we match more and more vertices, how do we ensure we do not get stuck? By always staying within the demand set of our domain thus far.

Proposition 9.1. *Suppose that we have a class $\{(A_\alpha, F_\alpha)\}$ where each A_α is a set of vertices (possibly infinite) and F_α is a matching from A_α into $D_G(A_\alpha)$. Then there is a matching F from $\bigcup A_\alpha$ into $\bigcup D_G(A_\alpha)$ (which, by the way, is contained in $D_G\left(\bigcup A_\alpha\right)$).*

Proof. Define F by matching each x to $F_\alpha(x)$, where α is least such that $x \in A_\alpha$. We show that F is a matching: suppose that $F(x_0) = y = F(x_1)$. Suppose that x_i first appears in A_{α_i} for $i = 0, 1$. Then $F_{\alpha_0}(x_0) = y = F_{\alpha_1}(x_1)$.

Since $F_{\alpha_0}(x_0) = y$, we have $y \in D_G(A_{\alpha_0})$. That means that $N_G(y) \subseteq A_{\alpha_0}$. Similarly, $N_G(y) \subseteq A_{\alpha_1}$. But $x_0, x_1 \in N_G(y)$, so $x_0, x_1 \in A_{\alpha_0} \cap A_{\alpha_1}$. It follows that x_0 and x_1 first appear in the same A_α (i.e., $\alpha_0 = \alpha_1$). Since F_α is a matching, we conclude that $x_0 = x_1$ as desired. $\square$

This gives us a way to implement step (1) in our strategy: simply combine all pairs (A, F) such that $A \subseteq X$ and F is a matching from A into $D_G(A)$. This gives us a matching F^* from A^* into $D_G(A^*)$.

We move on to step (2), where we have to construct a matching from $Y^* := Y - D_G(A^*)$ into $X^* := X - A^*$. We will do a clever inductive construction preserving the following property:

Proposition 9.2. *For all $A \subseteq X$ and all matchings $F : A \to D_G(A)$, every $y \in D_G(A) \cap Y^*$ is matched by F.*

Proof. By maximality of A^*, we have $A \subseteq A^*$. Therefore, $D_G(A) \subseteq D_G(A^*)$ which implies that $D_G(A) \cap Y^*$ is empty. □

In the following, we will consider (induced) subgraphs G' of G obtained by removing finitely many vertices from Y^* and X^*. We will denote the sides of G' by X' and Y'. Note that for all such G', we have $A^* \subseteq X'$ and $D_G(A^*) \subseteq Y'$.

Definition 9.3. We say that G' is *good* if for all $A \subseteq X'$ and all matchings $F : A \to D_{G'}(A)$, every $y \in D_{G'}(A) \cap Y^*$ is matched by F.

The previous proposition states that G is good. The definition of goodness and our matching F^* was carefully chosen to have the following combinatorial property:

Lemma 9.4. *Suppose that G' is good. Then for all $y \in Y' \cap Y^*$, there is some $x \in X^* \cap N_{G'}(y)$ such that $G' - \{x, y\}$ is still good.*

Now we may construct the desired matching from Y^* into X^* by repeatedly applying Lemma 9.4. To prove König's duality theorem, it remains to prove Lemma 9.4.

The following sub-lemma suffices to prove Lemma 9.4.

Lemma 9.5. *Suppose G' is good and $x \in X' \cap X^*$ and $y \in Y' \cap Y^*$ are such that $G' - \{x, y\}$ is not good. Then there is $A' \subseteq X'$ containing x, and a matching F' from A' into $D_{G'}(A')$ which leaves y unmatched.*

Proof of Lemma 9.4 using Lemma 9.5. We prove the contrapositive. Suppose that there is $y \in Y' \cap Y^*$ such that for all $x \in X^* \cap N_{G'}(y)$, $G' - \{x, y\}$ is not good. We show that G' is not good.

First, we claim that for all $x \in N_{G'}(y)$, there is a pair (A_x, F_x) such that $A_x \subseteq X'$, A_x contains x, and F_x is a matching from A_x into $D_{G'}(A_x)$ which leaves y unmatched.

If $x \in X^*$, then this is exactly the conclusion of Lemma 9.5. On the other hand, if $x \in X' - X^* \subseteq A^*$, then by definition of A^*, there is a pair (A_x, F_x) such that $A_x \subseteq A^*$, A_x contains x, and F_x is a matching from A_x into $D_G(A_x)$. We show that (A_x, F_x) satisfies the desired properties.

First note that $A_x \subseteq A^* \subseteq X'$. Also, $D_G(A_x) \subseteq D_G(A^*) \subseteq Y'$. Therefore $D_G(A_x) \subseteq D_{G'}(A^*)$. Finally, we note that y is unmatched:

this is because the range of F_x is contained in $D_G(A_x) \subseteq D_G(A^*)$, but y lies in $Y^* = Y - D_G(A^*)$.

This completes the proof of the claim. Now, use Proposition 9.1 to combine (A_x, F_x) for all $x \in N_{G'}(y)$ into some (A, F). Then (A, F) witnesses that G' is not good: it is easy to see that $A \subseteq X'$, the range of F lies in $D_{G'}(A)$, and that y is not matched by F. One only needs to check that $y \in D_{G'}(A)$, which holds because A contains $N_{G'}(y)$. $\square$

Finally, we prove Lemma 9.5, which is where the actual combinatorics happens. We will use the machinery of alternating paths. For any graph G and any matching F in G, an F-*alternating path* is a finite path in G such that alternate edges lie in F. If both the start and end vertices of an F-alternating path are not matched by F, then the path is said to be F-*augmenting*.

If we have an augmenting path for a matching, then said matching can be expanded by taking its symmetric difference with the augmenting path. The new matching will match all of the vertices which were previously matched, plus the start and end vertices of the augmenting path.

Proof of Lemma 9.5. Since $G' - \{x, y\}$ is not good, there is $A \subseteq X' - \{x\}$, a matching F from A into $D_{G'-\{x,y\}}(A)$, and some $y^* \in D_{G'-\{x,y\}}(A) \cap Y^*$ which is not matched by F. We define $A' = A \cup \{x\}$. We have to construct a matching F' from A' into $D_{G'}(A')$ which leaves y unmatched.

Suppose that we have an F-alternating path in G' from y^* to x. Note that neither y^* nor x are matched by F. Let F' be the augmentation of F using such a path. Since y^* lies in $D_{G'-\{x,y\}}(A)$, the range of F' still lies in $D_{G'-\{x,y\}}(A)$, which is contained in $D_{G'}(A')$. Also, y remains unmatched by F'. Hence F' is as desired.

It remains to show that there is indeed an F-alternating path in G' from y^* to x. To prove this, let S be the set of all $x' \in X' - \{x\}$ which can be reached via an F-alternating path in $G' - \{x, y\}$ starting from y^*, and let T be the set of all $y' \in Y' - \{y\}$ which can be reached via an F-alternating path in $G' - \{x, y\}$ starting from y^*. Observe the following:

- $S \subseteq A$, because $y^* \in D_{G'-\{x,y\}}(A)$ and $\mathrm{range}(F) \subseteq D_{G'-\{x,y\}}(A)$;
- $y^* \in T \subseteq D_{G'-\{x,y\}}(S)$;
- $F \restriction S$ is a matching from S into T.

If T were to be a subset of $D_{G'}(S)$, then S, $F \restriction S$, and $y^* \in T \subseteq D_{G'}(S)$ would witness that G' is not good, contradicting our assumption. Fix $y' \in T - D_{G'}(S)$. Since $y' \in D_{G'-\{x,y\}}(S)$, it follows that y' is adjacent to x in G'. This allows us to construct the desired F-alternating path:

first take an F-alternating path in $G' - \{x, y\}$ from y^* to y', then extend the path by adding the edge from y' to x. $\square$

9.1. **Proof of KDT in Π^1_1-CA_0.** Aharoni, Magidor, Shore [2] showed how to carry out the above proof in Π^1_1-CA_0. There are three places in the proof which deserve our attention (the rest of the proof can be carried out in ACA_0):

(1) in step (1) of our strategy, obtaining the set of all pairs (A, F) such that $A \subseteq X$ and F is a matching from A into $D_G(A)$;
(2) in the proof of Lemma 9.4 using Lemma 9.5, obtaining the set of all pairs (A_x, F_x) satisfying certain conditions;
(3) in the proof of the theorem by induction on Lemma 9.4.

In (1), the immediate issue is that the set of all pairs (A, F) such that $A \subseteq X$ and F is a matching from A into $D_G(A)$ is uncountable. We can work around that. First use Π^1_1-CA_0 to define the set

$$A^* = \{x \in X : \text{there is } A \ni x \text{ and a matching } F : A \to N_G(A)\}.$$

Then use Σ^1_1-AC_0 to obtain some

$$\{(A_x, F_x) : x \in A^*, x \in A_x, F_x : A_x \to N_G(A_x)\},$$

which we may then combine to obtain $F^* : A^* \to N_G(A^*)$.

(2) is also an application of Σ^1_1-AC_0.

In (3), we can use Π^1_1-CA_0 to obtain the set

$$\{\{x_0, \dots, x_n, y_0, \dots, y_n\} : G - \{x_0, \dots, x_n, y_0, \dots, y_n\} \text{ is good}\}.$$

We then do recursion to construct a matching of Y^* into X^*.

9.2. **Countable coded ω-models, proof of KDT in ATR_0.**

Definition 9.6 (RCA_0)**.** A *countable coded ω-model* is a set $M \subseteq \mathbb{N}$, whose columns encode the second-order part of an $\mathcal{L}_2$-structure.

Recursive comprehension with parameter M ensures that the set of $\mathcal{L}_2$-sentences φ with parameters from M exists. For any such φ, we say that a valuation for φ is a Boolean function f on the set of $\mathcal{L}_2$-sentences ψ which are substitution instances of subformulas of φ, satisfying certain properties, such as $f(\forall X \psi(X)) = 1$ iff $f(\psi((M)_n)) = 1$ for all $n \in \mathbb{N}$.

We say that M *satisfies* φ if there exists a valuation f for φ such that $f(\varphi) = 1$.

ACA_0 suffices to ensure that valuations exist for every φ.

Theorem 9.7 (Simpson)**.** ATR_0 *proves that for any $G \subseteq \mathbb{N}$, there is a countable coded ω-model $\mathcal{M}$ satisfying Σ^1_1-AC_0, such that $G \in \mathcal{M}$.*

With the above result on countable coded ω-models of Σ^1_1-AC_0 on hand, we describe how to carry out the proof of König's duality theorem in ATR_0.

Working in ATR_0, take $\mathcal{M}$ as above which contains our countable bipartite graph G. Now, we will relativize the constructions in the previous proof and the notion of goodness to $\mathcal{M}$. To start off, consider the set of all pairs $(A, F) \in \mathcal{M}$ such that $A \subseteq X$ and F is a matching from A into $D_G(A)$. Magic happens: this set is definable from a code of $\mathcal{M}$ (we can quantify over columns of the code), and hence exists by ACA_0 (in the ambient theory of ATR_0!) Therefore, we can combine those (A, F) to obtain a matching F^* from A^* into $D_G(A^*)$. Of course, F^* is only maximal "relative to $\mathcal{M}$", but that will suffice for our purposes. This completes step (1) of our strategy. Next,

Definition 9.8. We say that G' is *$\mathcal{M}$-good* if for all $A \subseteq X'$ and all matchings $F : A \to D_{G'}(A)$ such that $(A, F) \in \mathcal{M}$, every $y \in D_{G'}(A) \cap Y^*$ is matched by F.

The maximality of F^* relative to $\mathcal{M}$ is enough to ensure that G is $\mathcal{M}$-good!

Lemma 9.9. *Suppose G' is $\mathcal{M}$-good and $x \in X' \cap X^*$ and $y \in Y' \cap Y^*$ are such that $G' - \{x, y\}$ is not $\mathcal{M}$-good. Then there is $A' \subseteq X'$ containing x, and a matching F' from A' into $D_{G'}(A')$ which leaves y unmatched, such that $(A', F') \in \mathcal{M}$.*

The proof proceeds as before, using alternating paths. $\mathcal{M}$ satisfies ACA_0, which is enough to ensure that the various sets in the proof lie in $\mathcal{M}$.

Lemma 9.10. *Suppose that G' is $\mathcal{M}$-good. Then for all $y \in Y' \cap Y^*$, there is some $x \in X^* \cap N_{G'}(y)$ such that $G' - \{x, y\}$ is still $\mathcal{M}$-good.*

As before, we prove the contrapositive of Lemma 9.10. Assume that there is $y \in Y' \cap Y^*$ such that for all $x \in X^* \cap N_{G'}(y)$, $G' - \{x, y\}$ is not $\mathcal{M}$-good. We may show that for each $x \in N_{G'}(y)$, there is $(A, F) \in \mathcal{M}$ such that $A \subseteq X'$ contains x and $F : A \to N_{G'}(A)$ leaves y unmatched.

We then use Σ^1_1-AC_0 in $\mathcal{M}$ to show that some set of choices $\{(A_x, F_x) : x \in N_{G'}(y)\}$ lies in $\mathcal{M}$. Combining them, we obtain some $F : A \to N_{G'}(A)$ in $\mathcal{M}$ which leaves y unmatched. This witnesses that G' is not $\mathcal{M}$-good.

Finally, we prove KDT by induction on Lemma 9.10. The set of G' which is $\mathcal{M}$-good is arithmetic (in $\mathcal{M}$), so ACA_0 suffices to complete the induction.

10. Reducing KDT to ATR_2

In this section, we prove that KDT is arithmetically Weihrauch reducible to ATR_2. Our basic strategy is to follow Simpson's proof of KDT in ATR_0.

Consider the following problem: given a set G, produce a countable coded ω-model of Σ^1_1-AC which contains G.

Theorem 10.1 (Goh)**.** *The above problem is arithmetically Weihrauch reducible to* ATR_2.

Definition 10.2. Let L be a linear ordering. $I \subseteq L$ is a *cut* if:

- I is downward-$<_L$-closed;
- I has no $<_L$-largest element;
- the complement of I has no $<_L$-least element.

$I \subseteq L$ is a *proper cut* if I is a proper subset of L.

The following result is extracted from the proof of Simpson [53, Lemma 1].

Lemma 10.3. *If* $\langle X_a \rangle_{a \in L}$ *is a jump hierarchy on* L *which does not compute any proper cut of* L *and* I *is a proper cut of* L*, then the countable coded* ω*-model* $\mathcal{M} = \{A : \exists a \in I (A \leq_T X_a)\}$ *satisfies* Σ^1_1-AC.

Proof. Suppose we are given an arithmetic predicate $\varphi(n, Y)$ which is an instance of Σ^1_1-AC in $\mathcal{M}$. For each n, we claim that there is some $<_L$-least $a_n \in I$ such that X_{a_n} computes a solution to $\varphi(n, \cdot)$. Fix $b \in L \backslash I$ and consider the set

$$S = \{a <_L b : X_a \text{ computes a solution to } \varphi(n, \cdot)\}.$$

Since $\mathcal{M}$ contains a solution to $\varphi(n, \cdot)$, S intersects I. Also, as long as we fix b small enough, S is computable in $\langle X_a \rangle_{a \in L}$. Hence $(L \upharpoonright b) \backslash S$ is also computable in $\langle X_a \rangle_{a \in L}$. It follows that $(L \upharpoonright b) \backslash S$ is not a proper cut in L.

There are two possibilities: either S has a $<_L$-least element, in which case we are done, or $(L \upharpoonright b) \backslash S$ has a $<_L$-largest element. The latter case cannot happen because there would then be a computable $<_L$-descending sequence, contradicting a theorem of Friedman which states that any linear ordering which supports a jump hierarchy has no hyperarithmetic descending sequence.

We conclude that for each n, there must be some $<_L$-least $a_n \in I$ such that X_{a_n} computes a solution to $\varphi(n, \cdot)$.

Next, since I is a proper cut, for any $a \in I$ and $b \in L \backslash I$, X_b computes every X_a-hyperarithmetic set. Therefore if $b \in L \backslash I$, then X_b computes $(a_n)_{n \in \omega}$.

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

References

[1] Ron Aharoni. König's duality theorem for infinite bipartite graphs. *J. London Math. Soc. (2)*, 29(1):1–12, 1984.

[2] Ron Aharoni, Menachem Magidor, and Richard A. Shore. On the strength of König's duality theorem for infinite bipartite graphs. *J. Combin. Theory Ser. B*, 54(2):257–290, 1992.

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

[4] Dwight R. Bean. Effective coloration. *J. Symbolic Logic*, 41(2):469–480, 1976.

[5] Vasco Brattka. Computability over topological structures. In *Computability and models*, Univ. Ser. Math., pages 93–136. Kluwer/Plenum, New York, 2003.

[6] Vasco Brattka, Matthew de Brecht, and Arno Pauly. Closed choice and a uniform low basis theorem. *Annals of Pure and Applied Logic*, 163:986–1008, 2012.

[7] Vasco Brattka and Guido Gherardi. Effective choice and boundedness principles in computable analysis. *The Bulletin of Symbolic Logic*, 17(1):73–117, 2011.

[8] Vasco Brattka and Guido Gherardi. Weihrauch degrees, omniscience principles and weak computability. *The Journal of Symbolic Logic*, 76(1):143–176, 2011.

[9] Vasco Brattka and Guido Gherardi. Weihrauch goes Brouwerian. arXiv 1809.00380, 2018.

[10] Vasco Brattka, Guido Gherardi, and Alberto Marcone. The Bolzano-Weierstrass theorem is the jump of weak Kőnig's lemma. *Annals of Pure and Applied Logic*, 163:623–655, 2012.

[11] Vasco Brattka, Guido Gherardi, and Arno Pauly. Weihrauch complexity in computable analysis. arXiv 1707.03202, 2017.

[12] Vasco Brattka, Matthew Hendtlass, and Alexander P. Kreuzer. On the uniform computational content of computability theory. *Theory of Computing Systems*, 61(4):1376–1426, 2017.

[13] Vasco Brattka and Arno Pauly. On the algebraic structure of Weihrauch degrees. *Logical Methods in Computer Science*, 14(4:4):1–36, 2018.

[14] Vasco Brattka and Gero Presser. Computability on subsets of metric spaces. volume 305, pages 43–76. 2003. Topology in computer science (SchloßDagstuhl, 2000).

[15] Vasco Brattka and Tahina Rakotoniaina. On the uniform computational content of Ramsey's theorem. *Journal of Symbolic Logic*, 82(4):1278–1316, 2017.

[16] Keh Hsun Chen. Recursive well-founded orderings. *Ann. Math. Logic*, 13(2):117–147, 1978.

[17] Peter A. Cholak, Carl G. Jockusch, and Theodore A. Slaman. On the strength of Ramsey's theorem for pairs. *J. Symbolic Logic*, 66(1):1–55, 2001.

[18] Chi Tat Chong and Liang Yu. *Recursion theory*, volume 8 of *De Gruyter Series in Logic and its Applications*. De Gruyter, Berlin, 2015. Computational aspects of definability, With an interview with Gerald E. Sacks.

[19] Caleb Davis, Denis R. Hirschfeldt, Jeffry L. Hirst, Jake Pardo, Arno Pauly, and Keita Yokoyama. Combinatorial principles equivalent to weak induction. arXiv 1812.09943, 2018.

[20] Hannes Diener. Constructive Reverse Mathematics. arXiv 1804.05495, 2018.

[21] François G. Dorais, Damir D. Dzhafarov, Jeffry L. Hirst, Joseph R. Mileti, and Paul Shafer. On uniform relationships between combinatorial problems. *Transactions of the American Mathematical Society*, 368(2):1321–1359, 2016.

[22] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic randomness and complexity*. Theory and Applications of Computability. Springer, New York, 2010.

[23] Damir D. Dzhafarov. Cohesive avoidance and strong reductions. *Proceedings of the American Mathematical Society*, 143(2):869–876, 2015.

[24] Damir D. Dzhafarov. Strong reductions between combinatorial principles. *Journal of Symbolic Logic*, 81(4):1405–1431, 2016.

[25] Damir D. Dzhafarov. Joins in the strong Weihrauch degrees. *Mathematical Research Letters*, (to appear), 2018.

[26] Damir D. Dzhafarov, Jun Le Goh, Denis R. Hirschfeldt, Ludovic Patey, and Arno Pauly. Ramsey's theorem and products in the Weihrauch degrees. arXiv 1804.10968, 2018.

[27] Damir D. Dzhafarov and Carl G. Jockusch, Jr. Ramsey's theorem and cone avoidance. *J. Symbolic Logic*, 74(2):557–578, 2009.

[28] 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.

[29] Harvey Friedman. Some systems of second order arithmetic and their use. *Proceedings of the International Congress of Mathematicians (Vancouver, B. C., 1974), Vol. 1*, pages 235–242, 1975.

[30] Harvey M. Friedman and Jeffry L. Hirst. Weak comparability of well orderings and reverse mathematics. *Ann. Pure Appl. Logic*, 47(1):11–29, 1990.

[31] Peter Gerdes. *Moduli of computation*. PhD thesis, University of California, Berkeley, 2008.

[32] Guido Gherardi and Alberto Marcone. How incomputable is the separable Hahn-Banach theorem? *Notre Dame Journal of Formal Logic*, 50(4):393–425, 2009.

[33] Noam Greenberg and Antonio Montalbán. Ranked structures and arithmetic transfinite recursion. *Trans. Amer. Math. Soc.*, 360(3):1265–1307, 2008.

[34] Kojiro Higuchi and Arno Pauly. The degree structure of Weihrauch reducibility. *Log. Methods Comput. Sci.*, 9(2):2:02, 17, 2013.

[35] Denis R. Hirschfeldt. *Slicing the truth*, volume 28 of *Lecture Notes Series. Institute for Mathematical Sciences. National University of Singapore*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2015. On the computable and reverse mathematics of combinatorial principles, Edited and with a foreword by Chitat Chong, Qi Feng, Theodore A. Slaman, W. Hugh Woodin and Yue Yang.

[36] Denis R. Hirschfeldt and Carl G. Jockusch. On notions of computability-theoretic reduction between Π^1_2 principles. *Journal of Mathematical Logic*, 16(1):1650002, 59, 2016.

[37] Jeffry L. Hirst and Carl Mummert. Using Ramsey's theorem once. *Archive for Mathematical Logic*, pages 1–10, 2019.

[38] Jeffry Lynn Hirst. *Combinatorics in Subsystems of Second Order Arithmetic*. ProQuest LLC, Ann Arbor, MI, 1987. Thesis (Ph.D.)–The Pennsylvania State University.

[39] C. G. Jockusch, Jr. and T. G. McLaughlin. Countable retracing functions and Π^0_2 predicates. *Pacific J. Math.*, 30:67–93, 1969.

[40] Carl Jockusch and Frank Stephan. A cohesive set which is not high. *Math. Logic Quart.*, 39(4):515–530, 1993.

[41] Carl G. Jockusch, Jr. Ramsey's theorem and recursion theory. *J. Symbolic Logic*, 37:268–280, 1972.

[42] Takayuki Kihara, Alberto Marcone, and Arno Pauly. Searching for an analogue of ATR in the Weihrauch lattice. arXiv 1812.01549, 2018.

[43] Ulrich Kohlenbach. Higher order reverse mathematics. In *Reverse mathematics 2001*, volume 21 of *Lect. Notes Log.*, pages 281–295. Assoc. Symbol. Logic, La Jolla, CA, 2005.

[44] Christoph Kreitz and Klaus Weihrauch. Theory of representations. *Theoret. Comput. Sci.*, 38(1):35–53, 1985.

[45] Yiannis N. Moschovakis. Hyperarithmetical sets. In *Martin Davis on computability, computational logic, and mathematical foundations*, volume 10 of *Outst. Contrib. Log.*, pages 107–149. Springer, Cham, 2016.

[46] Ludovic Patey. The weakness of being cohesive, thin or free in reverse mathematics. *Israel Journal of Mathematics*, 216:905–955, 2016.

[47] Arno Pauly. How incomputable is finding Nash equilibria? *Journal of Universal Computer Science*, 16(18):2686–2710, 2010.

[48] Arno Pauly. On the (semi)lattices induced by continuous reducibilities. *Mathematical Logic Quarterly*, 56(5):488–502, 2010.

[49] Arno Pauly. On the topological aspects of the theory of represented spaces. *Computability*, 5(2):159–180, 2016.

[50] Klaus-Peter Podewski and Karsten Steffens. Injective choice functions for countable families. *J. Combinatorial Theory Ser. B*, 21(1):40–46, 1976.

[51] Gerald E. Sacks. *Higher recursion theory*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1990.

[52] Richard A. Shore. On the strength of Fraïssé's conjecture. In *Logical methods (Ithaca, NY, 1992)*, volume 12 of *Progr. Comput. Sci. Appl. Logic*, pages 782–813. Birkhäuser Boston, Boston, MA, 1993.

[53] Stephen G. Simpson. On the strength of König's duality theorem for countable bipartite graphs. *J. Symbolic Logic*, 59(1):113–123, 1994.

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

[55] Theodore A. Slaman and Marcia Groszek. Moduli of computation. Talk at Logic, Computability and Randomness, 2007.

[56] Klaus Weihrauch. *Computable Analysis*. Springer, Berlin, 2000.