CARDINAL INVARIANTS, NON-LOWNESS CLASSES, AND WEIHRAUCH REDUCIBILITY

NOAM GREENBERG, RUTGER KUYPER, AND DAN TURETSKY

Abstract. We provide a survey of results using Weihrauch problems to find analogs between set theory and computability theory. In our treatment, we emphasize the role of morphisms in explaining these coincidences. We end with a discussion of the use of forcing to prove the nonexistence of morphisms.

Contents

1. Introduction

In light of the independence of the continuum hypothesis, set theorists searched for more nuanced ways of measuring the size of the continuum. Perhaps, for example, the number of real numbers (singletons) is large, but it doesn't take many null sets (or meagre sets) to cover the real line? Perhaps there are many functions from ω to ω, but if we're only interested in growth rates, we can dominate all functions with only a few? Thus are defined *cardinal characteristics of the continuum*, such as the *dominating number* $\mathfrak{d}$, the smallest size of a family of functions dominating all functions, and $\mathbf{cov}(\mathscr{M})$, the smallest size of a collection of meagre sets whose union is $\mathbb{R}$. Again, the precise values of these cardinals are independent, but we can say much about the relationship between them. For example, no matter the size of the continuum, or the particular values of the two cardinals, we always have $\mathbf{cov}(\mathscr{M}) \leqslant \mathfrak{d}$; but strict inequality is consistent.

A. Miller (unpublished) and Fremlin [16] have noticed that many cardinal characteristics, including the two mentioned above, can be defined in terms of the smallest size of a set of "solutions" to "instances" of a problem, namely a binary relation. Let $\texttt{Capture}(\mathscr{M})$ be the membership relation between $\mathbb{R}$ and $\mathscr{M}$, the collection of meagre sets. Then $\mathbf{cov}(\mathscr{M})$ is the smallest size of a set $F \subseteq \mathscr{M}$ such that every instance ($x \in \mathbb{R}$) has a solution (a meagre set $A \ni x$) in F. The dominating number $\mathfrak{d}$ is similarly obtained from the domination relation $\mathtt{Dom}$ between functions in ω^ω.

Further, ZFC proofs of inequalities usually arise from *morphisms* between the associated problems. For the example above, we write $\mathtt{Capture}(\mathscr{M}) \to \mathtt{Dom}$ to indicate that there are functions $\psi_{\mathtt{inst}} \colon \mathbb{R} \to \omega^\omega$ and $\psi_{\mathtt{sol}} \colon \omega^\omega \to \mathscr{M}$ such that for all $x \in \mathbb{R}$ and $g \in \omega^\omega$, if $\psi_{\mathtt{inst}}(x)\,\mathtt{Dom}\,g$ then $x\,\mathtt{Capture}(\mathscr{M})\,\psi_{\mathtt{sol}}(g)$ (see definition 2.5 below). This immediately gives $\mathbf{cov}(\mathscr{M}) \leqslant \mathfrak{d}$, as the $\psi_{\mathtt{sol}}$-image of a complete solution set for $\mathtt{Dom}$ is a complete solution set for $\mathtt{Capture}(\mathscr{M})$. A thorough treatment of cardinal characteristics using these concepts was given by Vojtáš [46]. The morphism template gives a clearer presentation of arguments in this area; it also refines the question, because the existence of *definable* morphisms can be discussed even in the context of CH (see for example [6, 13, 39]).

At a similar time, Weihrauch and his school [47, 8, 17] independently developed similar concepts in their study of computable analysis. Again thinking of binary relations as "problems" with instances and solutions, in the light of computability theory, a computable morphism from a problem A to another problem B can be considered a reduction: B has at least as much information as A, because any method of solving B can effectively give us a method of solving A. Given an instance a of A, we effectively translate to an instance $\psi_{\mathtt{inst}}(a)$ of B, solve this problem, and then translate (via $\psi_{\mathtt{sol}}$) to an A-solution for a. When the maps $\psi_{\mathtt{inst}}$ and $\psi_{\mathtt{sol}}$ are indeed computable, this is called *strong Weihrauch reducibility.* This approach for classification of problems has spread beyond computable analysis to study Π^1_2 sentences in the context of reverse mathematics [14].

Rupprecht [41] observed similarities between several arguments in computability, especially algorithmic randomness, and set theory. A good example is Terwijn and Zambella's [45] proof of the equivalence of computable traceability and lowness for Schnorr tests, and Bartoszynski's [2] characterisation of $\mathbf{cof}(\mathscr{N})$ in terms of slaloms (see section 3.5 below). Rupprecht realised that the binary relations used for cardinal characteristics can be used to define some familiar notions of strength of oracles in computability. The class associated with the problem (binary relation) A, which in this paper we denote by $\mathrm{NL}(A)$, is the collection of oracles which compute an instance for A that has no computable solution. Thus, $\mathrm{NL}(\mathtt{Dom})$ is the collection of hyperimmune degrees (not $\mathbf{0}$-dominated), whereas $\mathrm{NL}(\mathtt{Capture}(\mathscr{M}))$ is the collection of oracles which compute weakly 1-generic reals. The morphism from $\mathtt{Capture}(\mathscr{M})$ to $\mathtt{Dom}$, being computable, shows that every degree which computes a weakly 1-generic real is hyperimmune.

Unfortunately, Rupprechet's main body of work was confined to his thesis, and remains otherwise unpublished. Aware of only some of his results, Brendle, Brooke-Taylor, Ng and Nies [10] extended his work and for the first time in print, exhibited how to get both cardinal characteristics and highness classes from the same binary relations (Weihrauch problems). Kihara [24] continued their work, including the morphism machinery, while Kjos-Hanssen et al. [26] answered some of the questions left open in [10].

In this paper we survey the subject and provide a unified and simplified treatment. Many arguments in the literature are opaque, and so we show how to frame them using this template. In many cases a single, simpler argument gives two, and with the aid of duality usually four, results, in set theory and computability.

We also provide a way of utilising sequential composition, which has so far eluded computability theorists. This allows us to give the same unified treatment to results of Stephan and Yu [44] about lowness for weak 1-genericity and Greenberg and J. Miller [19], related to [5] about lowness for Kurtz randomness, where, for a change, the argument found in computability gives cleaner morphisms. Nonetheless, we emphasise that most results in this paper are, essentially, not original.

Consistency results in set theory and non-implication results in computability are usually obtained by related forcing notions, and so we end the paper with a brief discussion of forcing. We examine a few basic examples to show how they fit in this framework. Finally we suggest a general research programme that arises naturally from the template that we discuss in this paper.

2. Basics

2.1. Weihrauch problems and effective morphisms.

Definition 2.1. A *Weihrauch problem* is a triple $A = (A_{\texttt{inst}}, A_{\texttt{sol}}, A)$ where $A_{\texttt{inst}}, A_{\texttt{sol}} \subseteq \omega^\omega$ and A is a binary relation between $A_{\texttt{inst}}$ and $A_{\texttt{sol}}$, that is, $A \subseteq A_{\texttt{inst}} \times A_{\texttt{sol}}$. An *$A$-instance* is an element of $A_{\texttt{inst}}$. A *solution* for an instance a of A is b such that aAb.[1]

The sets $A_{\texttt{inst}}$ and $A_{\texttt{sol}}$ are allowed to be complicated. However we will make one assumption:

- Both $A_{\texttt{inst}}$ and $A_{\texttt{sol}}$ contain computable points.

Example 2.2. $\texttt{Dom}$ is the domination problem: an instance is a function $f \in \omega^\omega$; a solution is a function g dominating f, denoted $g \geqslant^* f$, that is, $(\forall^\infty n)\, g(n) \geqslant f(n)$.

Definition 2.3. A (total) function $F\colon \omega^\omega \to \omega^\omega$ is *hyperarithmetic piecewise computable* if there is a countable partition $\langle A_n \rangle$ of ω^ω into uniformly Δ^1_1 sets such that for all n, $F\restriction A_n$ is the restriction to A_n of a partial computable function, uniformly in n.

That is, uniformly in n we get a Δ^1_1 index for A_n and a partial computable index (a c.e. Turing functional) for a partial computable function G_n such that $A_n \subseteq \operatorname{dom} G_n$ and $F\restriction A_n = G_n\restriction A_n$. An important fact to note is that for all $x \in \omega^\omega$, $F(x) \leqslant_{\mathrm{T}} x$, albeit not quite uniformly.

Remark 2.4. Fix an effective list Φ_e of all partial computable maps from ω^ω to ω^ω. A map $F\colon \omega^\omega \to \omega^\omega$ is hyperarithmetic piecewise computable if and only if there is a computable ordinal α and a partial computable map $\theta\colon \omega^\omega \to \omega$ such that for all $x \in \omega^\omega$, $\theta(x^{(\alpha)})\downarrow$ and $F(x) = \Phi_{\theta(x^{(\alpha)})}(x)$. Here $x^{(\alpha)}$ is the α^{th} iteration of the Turing jump of x.

Definition 2.5. An *effective morphism* φ from a problem A to a problem B is a pair $\varphi_{\texttt{inst}}, \varphi_{\texttt{sol}}$ of hyperarithmetic piecewise computable functions such that:

- $\varphi_{\texttt{inst}}[A_{\texttt{inst}}] \subseteq B_{\texttt{inst}}$;
- $\varphi_{\texttt{sol}}[B_{\texttt{sol}}] \subseteq A_{\texttt{sol}}$; and

[1]Blass and Rupprecht use "challenges" and "responses", or "answers", for instances and solutions. Coskey et al. and Kihara call Weihrauch problems "Vojtáš triples". Blass calls them "relations"; Rupprecht calls them "debates". We adopt terminology used in reverse mathematics, which is more widespread.

- for all $a \in A_{\texttt{inst}}$, for every B-solution b of $\varphi_{\texttt{inst}}(a)$, $\varphi_{\texttt{sol}}(b)$ is an A-solution of a.

We write $A \to B$ if there is an effective morphism from A to B.

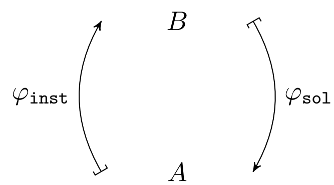

FIGURE 1. A morphism

The idea is that we are reducing A to B; given an A-instance a, we effectively translate it to a B-instance $\varphi_{\texttt{inst}}(a)$; from any B-solution b for $\varphi_{\texttt{inst}}(a)$ we get a solution $\varphi_{\texttt{sol}}(b)$ to the original instance. This is like strong Weihrauch reducibility, in that b computes $\varphi_{\texttt{inst}}(b)$ on its own, without aid of the original instance a; however unlike strong Weihrauch reductions, an effective morphism is not required to be uniform: it is almost uniform, except that we are allowed to define the reduction "by cases".[2]

Observation 2.6. If $A \to B$ and $B \to C$ then $A \to C$. Also $A \to A$. We write $A \leftrightarrow B$ when $A \to B$ and $B \to A$.

2.2. **The associated cardinal and highness class.**

Definition 2.7. Let A be a problem. A *complete solution set* for A is a set $Z \subseteq A_{\texttt{sol}}$ such that every A-instance has a solution in Z. We let

$$\mathrm{Card}(A) = \min\{|Z| \,:\, Z \text{ is a complete solution set for } A\}.$$ [3]

Definition 2.8. For a problem A we let $\mathrm{H}(A)$, the highness class associated with A, be the collection of oracles $x \in 2^\omega$ which compute some $c \in A_{\texttt{sol}}$ that solves every computable A-instance.

Example 2.9. A complete solution set for $\texttt{Dom}$ is a set of functions dominating all functions $f\colon \omega \to \omega$. $\mathrm{Card}(\texttt{Dom})$, denoted by $\mathfrak{d}$ and known as the *dominating number*, is the smallest size of a dominating family, that is, the cofinality of the pre-partial ordering $(\omega^\omega, \leqslant^*)$.

$\mathrm{H}(\texttt{Dom})$ is the collection of high degrees (Martin [30]): those that compute functions that dominate all computable ones.

Proposition 2.10 (Vojtáš [46]; Rupprecht [41])**.** *Suppose that $A \to B$. Then:*

(a) $\mathrm{Card}(A) \leqslant \mathrm{Card}(B)$*; and*

---

[2]Rupprecht uses completely non-uniform maps that only require $f(x) \leqslant_{\mathrm{T}} x$ for all x, anlogous to Muchnik reductions. Weihrauch and his school use the language of multi-valued functions rather than binary relations. Thus instead of the diagram in fig. 1 they draw a composition: $\varphi_{\texttt{sol}} \circ B \circ \varphi_{\texttt{inst}}(a) \subseteq A(a)$. The terminology for morphisms varies; Vojtáš called them "generalised Galois-Tukey connections"; Kihara "Tukey connections"; Rupprecht and Blass simply "morphism", but they reverse the direction.

[3]Blass and Rupprecht use the notation $||A||$ and call it the *norm* of A. Brendle et al. write $\mathfrak{d}(A)$ and call it the *domination number* of A.

(b) $\mathrm{H}(B) \subseteq \mathrm{H}(A)$.

Proof. Let $\varphi = (\varphi_{\mathtt{inst}}, \varphi_{\mathtt{sol}})$ be an effective morphism from A to B.

(a): Let Z be a complete solution set for B; let $W = \varphi_{\mathtt{sol}}[Z]$. Then $|W| \leqslant |Z|$ and W is a complete solution set for A.

(b): Let $x \in \mathrm{H}(B)$; let $b \leqslant_{\mathrm{T}} x$ be a B-solution for all computable B-instances. Let a be a computable A-instance. Then $\varphi_{\mathtt{inst}}(a)$ is a computable B-instance; so $\varphi_{\mathtt{inst}}(a)\ B\ b$. It follows that $\varphi_{\mathtt{sol}}(b)$ is an A-solution for a. As $\varphi_{\mathtt{sol}}(b) \leqslant_{\mathrm{T}} b \leqslant_{\mathrm{T}} x$, it witnesses that $x \in \mathrm{H}(A)$. □

2.3. **Duality.**

Definition 2.11. For a problem B we define its *dual* $B^{\perp}$ to be $\not{B}$. That is, the relation on $B_{\mathtt{sol}} \times B_{\mathtt{inst}}$ defined by $xB^{\perp}y \iff \neg(yBx)$. We let $(B^{\perp})_{\mathtt{inst}} = B_{\mathtt{sol}}$ and $(B^{\perp})_{\mathtt{sol}} = B_{\mathtt{inst}}$.

Hence $(A^{\perp})^{\perp} = A$.

Lemma 2.12. If $A \to B$ then $B^{\perp} \to A^{\perp}$.

Proof. Suppose that φ is an effective morphism from A to B. Define a morphism ψ by letting $\psi_{\mathtt{inst}} = \varphi_{\mathtt{sol}}$ and $\psi_{\mathtt{sol}} = \varphi_{\mathtt{inst}}$. Then ψ reduces $B^{\perp}$ to $A^{\perp}$: if $b \in B_{\mathtt{sol}} = (B^{\perp})_{\mathtt{inst}}$ and $a \in A_{\mathtt{inst}} = (A^{\perp})_{\mathtt{sol}}$ is such that $\psi_{\mathtt{inst}}(b)\, A^{\perp}\, a$, that is, $a \not{A}\, \varphi_{\mathtt{sol}}(b)$, then it cannot be that $\varphi_{\mathtt{inst}}(a)\, B\, b$, so $b\, B^{\perp}\, \psi_{\mathtt{sol}}(a)$. □

Definition 2.13. For a problem A, the non-lowness class associated with A is $\mathrm{NL}(A) = \mathrm{H}(A^{\perp})$.

So $x \in \mathrm{NL}(A)$ if and only if x computes an A-instance which has no computable A-solution.[4]

Corollary 2.14. *If* $A \to B$ *then* $\mathrm{NL}(A) \subseteq \mathrm{NL}(B)$.

Example 2.15. The dual of `Dom` is `Esc`: a solution for $g \in \omega^{\omega}$ is a function escaping g, that is, not dominated by g.

Card(`Esc`) $= \mathfrak{b}$ is the *unbounding number*: the smallest size of an unbounded family.

H(`Esc`) $=$ NL(`Dom`) is the collection of hyperimmune degrees (not **0**-dominated): those that compute escaping functions, functions not dominated by any computable function. NL(`Esc`) $=$ H(`Dom`) is high.

The problem `Esc` reduces to `Dom`: map an `Esc`-instance to itself; map a `Dom`-solution g to $g+1$. Hence: $\mathfrak{b} \leqslant \mathfrak{d}$; and every high degree is hyperimmune.[5]

---

[4]We used "highness" for $\mathrm{H}(A)$ because of the analogy with high degrees and their coincidence with H(`Dom`). We defined it first because its definition is more straightforward. However, we will see in the rest of the paper that the operator $\mathrm{NL}(A)$ is better behaved, in particular with respect to sequential composition, and its relationship with forcing. Rupprecht calls $\mathrm{NL}(A)$ the *Turing norm* of A and denotes it by $\langle A \rangle$. Brendle et al. write $\mathcal{D}(A)$ for $\mathrm{NL}(A)$ and $\mathcal{B}(A)$ for $\mathrm{H}(A)$, which they do define first; this is analogous to their notation $\mathfrak{b}(A)$ for $\mathrm{Card}(A^{\perp})$. They call both $\mathcal{D}(A)$ and $\mathcal{B}(A)$ "highness properties". Kihara writes $[A]_{\Delta^0_1}$ for $\mathrm{NL}(A)$.

[5] Our notation $\mathrm{NL}(A)$, as is Brendle et al.'s, is motivated by this simplest example, `Dom` and `Esc`: NL(`Dom`) is a weaker property that H(`Dom`). However this example is misleading, as usually there is no reduction from A to $A^{\perp}$ or vice-versa. Possibly better notation would be $\mathrm{H}^{\perp}(A)$ for $\mathrm{NL}(A)$; however we find this typographically annoying.

Example 2.16. Let $\mathtt{Split}$ be the *splitting problem*: an instance is an infinite subset a of ω (viewed as an element of $2^\omega \subset \omega^\omega$); a solution is an infinite subset b of ω which *splits* a, meaning that both $a \cap b$ and $a \backslash b = a \cap b^\complement$ are infinite. (Note that in this example the sets of instances and solutions are not all of ω^ω but a Π^0_2 subset of ω^ω.)

Card($\mathtt{Split}$), the *splitting number*, denoted by $\mathfrak{s}$, is the smallest size of a set of subsets of ω which split every infinite set. Card($\mathtt{Split}^\perp$), denoted by $\mathfrak{r}$ and sometimes called the *reaping number*, is the smallest size of a collection of infinite subsets of ω for which no single infinite subset of ω splits them all.

H($\mathtt{Split}$) is the collection of oracles which compute a *bi-immune* set: a set with no infinite c.e. subset of it or of its complement. This is because every infinite c.e. set contains an infinite computable set.

NL($\mathtt{Split}$) is the collection of oracles which compute *r-cohesive* sets: infinite sets c such that for every computable set a, $c \subseteq^* a$ or $c \subseteq^* a^\complement$.

Lemma 2.17. $\mathtt{Split} \to \mathtt{Dom}$.

Proof. We define a morphism ψ. On the instance side, map an infinite set $a \subseteq \omega$ to its principal function p_a. On the solution side, the idea is that if g dominates p_a then from g we can get a partition of ω into intervals $\langle I_n \rangle$, each of which contains an element of a; from this we easily build a set that splits a. So formally, given $g \in \omega^\omega$, we define $\psi_{\mathtt{sol}}(g)$ as follows. Let $h(0) = 0$ and $h(n+1) = g(h(n)+1)$; let $I_n = (h(n), h(n+1)]$; let $\psi_{\mathtt{sol}}(g) = \bigcup_n I_{2n}$. □

As a result, we see that $\mathfrak{s} \leqslant \mathfrak{d}$, $\mathfrak{b} \leqslant \mathfrak{r}$, every high degree computes a bi-immune set, and every r-cohesive set has hyperimmune degree.

3. Represented spaces, measure and category

Since we allow the collections $A_{\mathtt{inst}}$ and $A_{\mathtt{sol}}$ of instances and solutions of a problem to be subsets of ω^ω, we can formulate problems whose instances or solutions, for example, are elements of Cantor space, or collections of bounded functions. However at times we will want to deal with objects that are not in Baire space, for example, real numbers, or some definable subsets of Baire space. For this, we use the terminology of represented spaces.

Definition 3.1. A *representation* of a set X is a partial function from ω^ω onto X.

If π is a representation of X then $a \in \operatorname{dom} \pi$ is called a π-*name* for $\pi(a)$. We then say that $y \in 2^\omega$ *computes* $x \in X$ if x has a y-computable name. We choose representations so that this aligns with our preconceived notions of computation.

Below we will define problems whose instances or solutions are elements of represented spaces. In all cases this is shorthand for the induced problems on the names: if π is a representation of X, for example, then we will define a problem $A \subseteq X \times \omega^\omega$, but really we will mean the problem $\hat{A} \subseteq \operatorname{dom} \pi \times \omega^\omega$ defined by $\hat{A}_{\mathtt{inst}} = \pi^{-1} A_{\mathtt{inst}}$ and $x\,\hat{A}\,y$ means $\pi(x)\,A\,y$. We will identify A with $\hat{A}$.

3.1. Meagre sets.

The first example is the collection of $\mathbf{\Sigma}^0_2$ meagre subsets of Baire space. A name for a closed, nowhere dense subset P of ω^ω is the characteristic function of a tree $T \subseteq \omega^{<\omega}$ such that $P = [T]$ (we use a computable bijection between ω and $\omega^{<\omega}$). Trees are allowed to have leaves (strings with no extensions on the tree). By a canonical identification of Baire space with its power $(\omega^\omega)^\omega$, we let $\langle T_n \rangle$ be a name for $\bigcup [T_n]$.

We let $\mathscr{M}$ be the collection of $\boldsymbol{\Sigma}^0_2$ meagre sets. As described above, we identify a name for a meagre set with the set that it names, so we also think of $\mathscr{M}$ as the collection of names for $\boldsymbol{\Sigma}^0_2$ meagre sets.

Remark 3.2. Since every effectively closed set is the set of paths through a computable tree, and this is uniform and relativises, we see that a $\boldsymbol{\Sigma}^0_2$ meagre set is x-computable if and only if it is $\Sigma^0_2(x)$.

Two problems are typically associated with ideals of small sets, in this case the meagre ones:

- $\texttt{Capture}(\mathscr{M})$: an instance is a point $y \in \omega^\omega$; a solution is a meagre set M such that $y \in M$.
- $\texttt{Supset}(\mathscr{M})$: an instance is a meagre set M; a solution is a meagre set $\tilde{M}$ such that $M \subseteq \tilde{M}$.

Card($\texttt{Capture}(\mathscr{M})$) is known as $\mathbf{cov}(\mathscr{M})$: the smallest number of meagre sets it takes to cover ω^ω. H($\texttt{Capture}(\mathscr{M})$) is the class of $x \in 2^\omega$ which compute a meagre set that contains all computable reals: the *weakly meagre engulfing* oracles.[6]

The dual of $\texttt{Capture}(\mathscr{M})$ is $\texttt{Pass}(\mathscr{M})$, the problem of finding a point outside a given meagre set. Card($\texttt{Pass}(\mathscr{M})$) is $\mathbf{non}(\mathscr{M})$ (also known as the *uniformity number* $\mathbf{unif}(\mathscr{M})$), the smallest size of a non-meagre set. H($\texttt{Pass}(\mathscr{M})$) is the class of $x \in 2^\omega$ which compute a weakly 1-generic function.

Card($\texttt{Supset}(\mathscr{M})$) is $\mathbf{cof}(\mathscr{M})$, the cofinality of the partial order $(\mathscr{M}, \subseteq)$. The class H($\texttt{Supset}(\mathscr{M})$) consists of the (strongly) *meagre englufing* oracles: those which compute a meagre set covering (i.e. a superset of) all computable meagre sets.

The dual of $\texttt{Supset}(\mathscr{M})$ is $\texttt{Spill}(\mathscr{M})$, the problem of finding a meagre set which is not covered by a given meagre set. Card($\texttt{Spill}(\mathscr{M})$) is $\mathbf{add}(\mathscr{M})$, the *additivity number* for the meagre sets: the smallest number of meagre sets whose union is not meagre. H($\texttt{Spill}(\mathscr{M})$) is the class of oracles which are not low for meagre sets.

Proposition 3.3. $\texttt{Capture}(\mathscr{M}) \to \texttt{Dom}$.

Proof. Map the instance f to itself. Map a $\texttt{Dom}$-solution $g \in \omega^\omega$ to the meagre set consisting of all functions dominated by g.

More precisely, let $T_n(g)$ be the tree of $\sigma \in \omega^{<\omega}$ such that for all $k \geqslant n$, if $k < |\sigma|$ then $\sigma(k) \leqslant g(k)$; map g to $\langle T_n(g) \rangle$. ☐

As a result we see:

- $\mathbf{cov}(\mathscr{M}) \leqslant \mathfrak{d}$ and $\mathfrak{b} \leqslant \mathbf{non}(\mathscr{M})$.
- Every high degree is weakly meagre engulfing; every weakly 1-generic degree is hyperimmune.

Proposition 3.4. $\texttt{Pass}(\mathscr{M}) \to \texttt{Supset}(\mathscr{M})$.

Proof. Uniformly, given (a name of) a meagre set M, we can find a point $x \notin M$: since M is given as a sequence $\langle T_n \rangle$ of nowhere dense trees, we construct $x \notin \bigcup_n [T_n]$ by initial segments, first finding a string σ_0 off T_0, then an extension σ_1 off T_1, and so on.

The morphism maps an instance M of $\texttt{Pass}(\mathscr{M})$ to itself, and a solution $\tilde{M} \supseteq M$ to a real outside $\tilde{M}$.

[6]The terminology originates from [10].

Note that for the map of solutions to be total, we cannot produce a partial sequence $\langle \sigma_n \rangle$ in case we are given a tree which is somewhere dense. However the collection of names for meagre sets is arithmetic, so we are allowed to define the map of solutions in an arbitrary way outside $\mathscr{M}$. □

Remark 3.5. The previous morphism exhibits the importance of allowing functions on names that do not induce maps on the named objects. Given two names $\langle T_n \rangle$ and $\langle S_n \rangle$ of the same meagre set M, the resulting point outside M will very likely depend on the name.

Proposition 3.6. *$\texttt{Dom} \to \texttt{Supset}(\mathscr{M})$.*

Proof. On the instance side, map a function f to the meagre set M_f consisting of all f-dominated functions. By first replacing f by $\sum_{i \leqslant n} f(i)$, we may assume that f is non-decreasing.

On the solution side, we elaborate on the construction of a point outside a given meagre set $M = \bigcup_n [T_n]$ (applying that construction would yield a morphism from $\texttt{Esc}$ to $\texttt{Supset}(\mathscr{M})$, which is weaker). Given $\langle T_n \rangle$, for each n and $k < \omega$ we can effectively obtain a string $\sigma_{n,k}$ such that for all k-bounded strings τ of length k, $\tau\hat{\ }\sigma_{n,k} \notin T_n$. We may assume that $T_0 \subseteq T_1 \subseteq T_2 \subseteq \dots$. By extending $\sigma_{n,k}$, we may assume that $\max \operatorname{range} \sigma_{n,k} > k$.

We then recursively define $g(0) = 0$ and $g(n+1) = \max \operatorname{range} \sigma_{n,g(n)}$. So g is strictly increasing. Let $J_n = [g(n), g(n) + |\sigma_{n,g(n)}|]$. For all $y \in \omega^\omega$, if $y \restriction g(n)$ is $g(n)$-bounded and $y \restriction J_n = \sigma_{n,g(n)}$, then $y \notin [T_n]$.

We claim that if $M_f \subseteq M$, that is, if M contains all f-dominated functions, then $f \leqslant g(n+1)$.

Suppose not. Then there is an infinite set X such that for all $n \in X$, $g(n+1) \leqslant f(n)$. We may thin out X so that the intervals J_n for $n \in X$ are pairwise disjoint and $m > n$ in X implies $m > g(n+1)$.

We can then build a function $x \in \omega^\omega$ such that for all $n \in X$, $x \restriction J_n = \sigma_{n,g(n)}$. Outside these intervals we set the values of x to be 0. If $n \in X$ and $i \in J_n$ then $x(i) = \sigma_{n,g(n)}(i) \leqslant g(n+1)$. If $m \in X$ and $m > n$ then as $m > g(n+1)$, this shows that $x \restriction g(m)$ is $g(m)$-bounded. It follows that $x \notin M$. On the other hand, for $n \in X$, $g(n+1) \leqslant f(n)$ and as f is non-decreasing, $f(n) \leqslant f(i)$ for all $i \in J_n$, so $x \leqslant f$. □

Combining propositions 3.3 and 3.6 we see that $\texttt{Capture}(\mathscr{M}) \to \texttt{Supset}(\mathscr{M})$ (and so also $\texttt{Spill}(\mathscr{M}) \to \texttt{Pass}(\mathscr{M})$). This actually has a quick direct proof: on the instance side, map a function f to the singleton $\{f\}$ (it is easy to effectively obtain a tree T such that $[T] = \{f\}$). On the solution side we use the identity function.

3.2. **Other spaces.** We have looked at meagre subsets of Baire space, but we could equivalently examine either Cantor space or the real line. We show that the corresponding problems are all morphism-equivalent. For this subsection, for any one of the spaces $X \in \{\omega^\omega, 2^\omega, \mathbb{R}\}$, denote by $\mathscr{M}_X$ the collection of $\mathbf{\Sigma}^0_2$ meagre subsets of X.

Let us first deal with Cantor space. Here $\mathscr{M}_{2^\omega}$ is a subset of $\mathscr{M}_{\omega^\omega}$, with the names being infinite sequences of nowhere dense subtrees of $2^{<\omega}$. Define $\rho\colon \omega^\omega \to 2^\omega$ by letting

$$\rho(f) = 0^{f(0)}10^{f(1)}10^{f(2)}1\cdots;$$

It is a bijection between Baire space and the collection of (characteristic functions of the) infinite subsets of ω. The map ρ is induced by a corresponding map from $\omega^{<\omega}$ to $2^{<\omega}$, which we also call ρ: $\rho(\langle n_0, n_1, \dots, n_k\rangle) = 0^{n_0}10^{n_1}1\cdots 10^{n_k}$.

For any meagre set $M \subset 2^\omega$, $\rho^*(M) = \rho^{-1}[M]$ is a meagre subset of ω^ω. Further, a name for $\rho^*(M)$ can be effectively obtained from a name for M: the sequence $\langle T_n\rangle$ (with $T_n \subset 2^{<\omega}$) is sent to $\langle \rho^{-1}[T_n]\rangle$.

For $M \subseteq \omega^\omega$ we let $\rho_*(M) = \rho[M] \cup \{x \in 2^\omega : x \text{ is finite}\}$. M is meagre if and only if $\rho_*(M)$ is meagre, and a name for $\rho_*(M)$ is effectively obtained from a name for M: map a sequence $\langle S_n\rangle$ to $\langle T_n\rangle$ where

$$T_n = \left\{\rho(\sigma)\hat{\ }0^k \,:\, \sigma \in S_n \;\&\; k < \omega\right\} \cup \{\tau \in 2^{<\omega} \,:\, \tau \text{ has up to } n \text{ 1's}\}.$$

Lemma 3.7. $\texttt{Capture}(\mathscr{M}_{2^\omega}) \leftrightarrow \texttt{Capture}(\mathscr{M}_{\omega^\omega})$ and $\texttt{Supset}(\mathscr{M}_{2^\omega}) \leftrightarrow \texttt{Supset}(\mathscr{M}_{\omega^\omega})$.

Proof. To reduce $\texttt{Capture}(\mathscr{M}_{\omega^\omega})$ to $\texttt{Capture}(\mathscr{M}_{2^\omega})$, on the instance side use ρ and on the solution side use ρ^*. In the other direction, on the instance side map an infinite set x to $\rho^{-1}(x)$, and a finite set x to some fixed point, say the constant function 0. On the solution side use ρ_*. In either case this will work, because $\rho_*(M)$ will always contain all finite sets.

To reduce $\texttt{Supset}(\mathscr{M}_{\omega^\omega})$ to $\texttt{Supset}(\mathscr{M}_{2^\omega})$, use ρ_* on the instance side and ρ^* on the solution side. We just need to check that if $\tilde{M} \supseteq \rho_*(M)$ then $M \subseteq \rho^*(\tilde{M})$. In the other direction, use ρ^* on the instance side and ρ_* on the solution side; check the same thing. ☐

We conclude that $\mathbf{cov}(\mathscr{M}_{2^\omega}) = \mathbf{cov}(\mathscr{M}_{\omega^\omega})$, $\mathbf{cof}(\mathscr{M}_{2^\omega}) = \mathbf{cof}(\mathscr{M}_{\omega^\omega})$ (and the same holds for $\mathbf{add}(\mathscr{M})$ and $\mathbf{non}(\mathscr{M})$), and that the highness classes coincide. So: computing a weakly 1-generic function in Baire space is equivalent to computing a weakly 1-generic sequence in Cantor space, lowness for meagre sets coincides, and the engulfing properties (weak and strong) coincide for Baire space and Cantor space.

For the real numbers, we need to choose representations. We use standard ones; $\mathbb{R}$ is represented using Cauchy names (quickly converging Cauchy sequence) — although since we do not require the morphism maps to be uniform, representation using binary expansions would be equivalent. Open subsets of $\mathbb{R}$ are represented by sequences of rational intervals (named by their endpoints) whose union is the open set; closed names are given by taking the complement; names for $\mathbf{\Sigma}^0_2$ subsets of $\mathbb{R}$ are sequences of closed names. As above, we can use Baire category-preserving maps to obtain morphism equivalence of the problems $\texttt{Capture}(\mathscr{M}_{\omega^\omega})$ and $\texttt{Capture}(\mathscr{M}_{\mathbb{R}})$, usually passing through the unit interval first. For example, we can use the binary expansion map from Cantor space onto $[0,1]$, or the homeomorphism of Baire space with the irrationals in $(0,1)$. We just need to check that these maps are Σ^0_2-measurable, effectively.

Remark 3.8. The reduction $\texttt{Capture}(\mathscr{M}_{2^\omega}) \to \texttt{Capture}(\mathscr{M}_{\omega^\omega})$ is an instance of a morphism ψ for which $\psi_{\texttt{inst}}\restriction A_{\texttt{inst}}$ is not computable (the restriction of a partial computable function to $A_{\texttt{inst}}$). This is because the map depends on whether x is infinite or not. This, and the reduction for $\mathscr{M}_{\mathbb{R}}$, are the only such reductions in this paper.

3.3. **Null sets.** We use similar techniques to name null $\mathbf{\Pi}^0_2$ subsets of 2^ω. An open set is named by (the characteristic function of) a subset U of $2^{<\omega}$; we may assume it is upwards closed in $2^{<\omega}$. This means that under this naming scheme, x is a name for an open set U if and only if the complement $x^{\complement}$ of x is a name for the closed set $2^\omega \backslash U$ under the scheme of naming closed sets by trees used above. We abuse notation by using U to denote both a set of strings and the open set (sometimes denoted $[U]^{\prec}$) generated by U.

As we are dealing with relatively computable names, rather than relatively c.e. names, we do not require that $[\sigma] \subseteq U$ implies $\sigma \in U$. This is the analogue of allowing terminal nodes in our trees naming closed sets.

A name for a null $\mathbf{\Pi}^0_2$ set is essentially a Schnorr test: it consists of a pair of sequences $\langle U_n \rangle$ and $\langle \lambda(U_n) \rangle$, where U_n is (a name for) an open set and $\lambda(U_n)$ is (a name for) the real number which is the fair-coin measure of U_n. We require that $\lambda(U_n) \leqslant 2^{-n}$. The null set named is $\bigcap_n U_n$. We do not require the test to be nested, but this can be obtained by replacing $\langle U_n \rangle$ by $\langle V_n \rangle$ where $V_n = \bigcup_{m>n} U_m$; the sequence $\langle \lambda(V_n) \rangle$ (as well as the sequence of sets of strings $\langle V_n \rangle$) is computed from $\langle U_n \rangle$ and $\langle \lambda(U_n) \rangle$.

We note that an alternative naming system would be omitting the sequence of measures $\langle \lambda(U_n) \rangle$ and simply requiring that $\lambda(U_n) = 2^{-n}$ (see for example [15, Prop.7.1.6]). We also remark that unlike the naming scheme for $\mathbb{R}$, when naming infinite sequences of real numbers we do need to use quickly converging Cauchy sequences rather than binary expansions, as we cannot pass between them using piecewise continuous functions: for each coordinate we need to know whether the real in that coordinate is a binary rational or not, and there are continuum many possibilities.

For any oracle x, the x-computable null sets are the x-Schnorr null sets. The associated cardinals and highness classes are:

- Card($\texttt{Capture}(\mathscr{N})$) = $\mathbf{cov}(\mathscr{N})$ is the smallest number of null sets it takes to cover the reals; the elements of H($\texttt{Capture}(\mathscr{N})$) are the *weakly null engulfing* (or weakly Schnorr engulfing) oracles, those that compute a null set containing all computable points.
- Card($\texttt{Supset}(\mathscr{N})$) = $\mathbf{cof}(\mathscr{N})$ is the cofinality of $(\mathscr{N}, \subseteq)$; H($\texttt{Supset}(\mathscr{N})$) is the class of (strongly) *null engulfing* oracles, those that compute a null set covering all computable Schnorr null sets.
- Card($\texttt{Pass}(\mathscr{N})$) = $\mathbf{non}(\mathscr{N})$ is the smallest size of a non-null set; an oracle is in H($\texttt{Pass}(\mathscr{N})$) if and only if it computes a Schnorr random real.
- Card($\texttt{Spill}(\mathscr{N})$) = $\mathbf{add}(\mathscr{N})$ is the smallest number of null sets whose union is not null; H($\texttt{Spill}(\mathscr{N})$) is the class of oracles which are not low for Schnorr null sets.

The two basic morphisms that apply to most σ-ideals apply to null sets as well:

Proposition 3.9. $\texttt{Capture}(\mathscr{N}) \to \texttt{Supset}(\mathscr{N})$ *and* $\texttt{Pass}(\mathscr{N}) \to \texttt{Supset}(\mathscr{N})$.

Proof. The same as for meagre, noting that given a name $\langle U_n \rangle$ for a null set we can effectively compute a point outside the null set $\bigcap_n U_n$. Indeed we can build $x \notin U_1$, defining it bit by bit, by ensuring that $\lambda([x\upharpoonright n] \backslash U_1) > 0$. We use the fact that for all clopen C, $\lambda(C \cap U_1)$ is uniformly computable from U_1 and $\lambda(U_1)$.

Also, it is easy, given $x \in 2^\omega$, to find a name for $\{x\}$ as a null set. □

Remark 3.10. By forgetting the sequence $\langle \lambda(U_n) \rangle$, we could have easily defined the representation of null sets using Martin-Löf tests, which would make the effective notions correspond to computing ML-random reals and so on. But the most basic morphisms (as in proposition 3.9) would not work for such a representation. Schnorr is the correct effective analogue to set-theoretic randomness, because in set theory there is no sense to "not knowing" the exact measure of an open set.

3.4. **Category and measure – an easy morphism.** We start by observing a relatively simple connection between category and measure. We use:

Lemma 3.11. There are an effectively null (Schnorr null) set and an effectively meagre set which form a partition of 2^ω.

Proof. This can be done in many ways. For example, use a pairing function to identify ω with ω^2, and let $U_n = \bigcup_m C_{n,k}$, where $C_{n,k}$ is determined by finitely many bits on the k^{th} column of ω; for example

$$C_{n,k} = \{x \in 2^\omega \,:\, x(k,0) = x(k,1) = \cdots = x(k,k+n) = 0\}\,. \qquad \square$$

Proposition 3.12. $\texttt{Capture}(\mathscr{M}) \to \texttt{Pass}(\mathscr{N})$.

Dualising we get $\texttt{Capture}(\mathscr{N}) \to \texttt{Pass}(\mathscr{M})$. As a result, $\mathbf{cov}(\mathscr{M}) \leqslant \mathbf{non}(\mathscr{N})$ and $\mathbf{cov}(\mathscr{N}) \leqslant \mathbf{non}(\mathscr{M})$; and every Schnorr random is weakly meagre engulfing, whereas every weakly 1-generic is weakly null engulfing.

Proof. By lemma 3.11, we fix a computable null set N and a computable meagre set M which partition 2^ω. We then let, for $x \in 2^\omega$,

$$\psi_{\texttt{inst}}(x) = x \triangle N = \{x \triangle y \,:\, y \in N\}$$

and for $y \in 2^\omega$ we let $\psi_{\texttt{sol}}(y) = y \triangle M$. The map $x \mapsto x \triangle y$ is measure and category-invariant, so indeed $x \triangle N$ is null and $y \triangle M$ is meagre. And if $y \notin x \triangle N$ then $x \triangle y \notin N$ so $x \triangle y \in M$ so $x \in y \triangle M$. $\square$

3.5. **Tracing.** The material in this section was discovered by Bartoszynski [2], by Raisonnier and Stern [40], and then independently by Terwijn and Zambella [45].

A *trace* is a function $T\colon \omega \to \mathscr{P}_\omega(\omega) = [\omega]^{<\omega}$. An *order function* is a computable, non-decreasing, unbounded function.[7] If h is an order function then an h-*trace* is a trace T such that for all n, $|T(n)| \leqslant h(n)$. In set theory, traces are known as *slaloms*. A trace T *traces* a function $f \in \omega^\omega$ if $(\forall^\infty n)\, f(n) \in T(n)$. By canonically coding finite sets of natural numbers by natural numbers, we represent traces by elements of Baire space. The most basic problem involving traces is h-$\texttt{Trace}$: an instance is a function $f \in \omega^\omega$, a solution is an h-trace T which traces f.

Traces are used extensively in algorithmic randomness and computability. See, for example, [15, 36, 21].

Lemma 3.13. For any two order functions h and $\tilde{h}$, h-$\texttt{Trace} \leftrightarrow \tilde{h}$-$\texttt{Trace}$.

We therefore simply write $\texttt{Trace}$ for the equivalent problems h-$\texttt{Trace}$.

Proof. Let h, $\tilde{h}$ be order functions; we reduce h-$\texttt{Trace}$ to $\tilde{h}$-$\texttt{Trace}$. We say that k is responsible for n if $\tilde{h}(k) \leqslant h(n) < \tilde{h}(k+1)$. Then for every n with $h(n) \geqslant \tilde{h}(0)$, a unique k is responsible for n, and every k is responsible for finitely many n.

On the instance side, map $f \in \omega^\omega$ to $\tilde{f}$ defined by $\tilde{f}(k) = f\upharpoonright(n+1)$, where n is the greatest such that k is responsible for n. If k is responsible for no n (for example if $\tilde{h}(k) = \tilde{h}(k+1)$) then it doesn't matter what $\tilde{f}(k)$ is.

On the solution side, we map a $\tilde{h}$-trace T to the h-trace S defined by

$$S(n) = \{\sigma(n) \,:\, \sigma \in T(k),\ \text{where } k \text{ is responsible for } n\}\,,$$

[7] Terminology by Schnorr [43].

letting $S(n)$ be empty if $h(n) < \tilde{h}(0)$. ☐

NL(`Trace`) is the collection of oracles which are not *computably traceable*. An oracle x is computably traceable if every x-computable function has a computable h-trace, for some fixed (or all) order functions h. It is not difficult to see that `Dom` $\to$ `Trace` (an instance is mapped to itself, a trace S is mapped to $n \mapsto \max S(n)$); the resulting containment NL(`Dom`) $\subseteq$ NL(`Trace`) says (in contrapositive) that every computably traceable degree is **0**-dominated (hyperimmune-free). The containment H(`Trace`) $\subseteq$ H(`Dom`) says that if there is an x-computable h-trace that traces all computable functions, then x is high. In fact, it is not difficult to see that H(`Trace`) = H(`Dom`), that is, H(`Trace`) is the collection of high degrees: suppose that x is high. The set of indices of computable functions is $\Delta^0_2(x)$. Approximating this set, we let $T(n)$ be the set of values $\varphi_{e,s}(s)$ for all $e < s$ which are believed to be total at stage s. This is an identity-bounded, x-computable trace that traces all computable functions. Note how this proof relies on special properties of high degrees (and the enumeration of partial computable functions) that go beyond merely using a dominating function. This proof has no analogue in set theory, and indeed, the associated cardinals can be distinct.

Our next goal is the following combinatorial characterisation of the cofinality of the null ideal.

Theorem 3.14. `Trace` $\leftrightarrow$ `Supset`$(\mathscr{N})$.

Proof of `Supset`$(\mathscr{N}) \to$ `Trace`. Define an order function h so that for large enough n, $2^{-2n}\sum_{k<\omega} h(2n,k)\cdot 2^{-k} \leqslant 2^{-n}$, and that this sum is computable, uniformly in n.

On the instance side: given a null set $N = \bigcap U_n$, let, for $n, k < \omega$, $s(n,k)$ be a stage s such that $\lambda(U_{n,s}) \geqslant \lambda(U_n)\cdot(1-2^{-k})$, where $U_{n,s} = U_n \cap \{0,1\}^{\leqslant s}$. Let $s(n,-1) = 0$. We map N to the function $(n,k) \mapsto U_{n,s(n,k)} - U_{n,s(n,k-1)}$. A value of this function is a clopen set, so is coded by a finite subset of $2^{<\omega}$, and hence by a natural number.

On the solution side, given an h-trace T, we may assume that for all n, k, every element of $T(n,k)$ is a clopen set of measure at most 2^{-n-k}. We then map T to a name $\langle V_n\rangle$ for a null set, defined by letting, for sufficiently large n, $V_n = \bigcup_{k<\omega}\bigcup T(2n,k)$. The properties of h ensure that $\lambda(V_n) \leqslant 2^{-n}$. The uniform computability of the sums above implies that the sequence $\langle\lambda(V_n)\rangle$ can be obtained from T.

For all n, $U_{2n} \subseteq V_n$, so $\bigcap U_n \subseteq \bigcap V_n$. ☐

Proof of `Trace` $\to$ `Supset`$(\mathscr{N})$. We define an array of clopen sets $B_{m,k}$:

$$B_{m,k} = \{x \in 2^\omega \,:\, x(k,0) = x(k,1) = \cdots = x(k,m-1) = 0\}\,.$$

For each m, the family of sets $\langle B_{m,k}\rangle$ is independent (in the sense of probability theory), and $\lambda(B_{m,k}) = 2^{-m}$. Further, for any clopen set C and m, for some k, the family $\{C, B_{m,k}, B_{m,k+1}, B_{m,k+2}, \dots\}$ is independent.

We also fix a computable map $D \mapsto q_D$ from clopen sets to positive binary rational numbers, such that $\sum_D q_D \leqslant 1/8$. For each clopen set D, we let $h_D(n) = \lceil 1/\log_{q_D}(1-2^{-n})\rceil$. This is an order function. The *granularity* of a clopen set D is the least k such that D is the union of clopen sets $[\sigma]$ for binary strings σ of length k. We let

$$h(n) = \sum\{h_D(n) \,:\, D \text{ is a clopen set of granularity} \leqslant n\}\,.$$

We reduce h-Trace to $\texttt{Supset}(\mathscr{N})$.

On the instance side, the map is relatively simple: we map $f \in \omega^\omega$ to $\langle U_n \rangle$, where $U_n = \bigcup_{m>n} B_{m,f(m)}$. The sequence $\langle \lambda(U_n) \rangle$ is obtained effectively from f.

On the solution side, we work a bit. Given a name $\langle V_n \rangle, \langle \lambda(V_n) \rangle$ of a null set, we will actually only use V_2 and $\lambda(V_2)$. We first define an open set $W \supseteq V_2$ such that $\lambda(W) \leqslant 1/2$, W and $\lambda(W)$ are computable given V_2 and $\lambda(V_2)$, and for every clopen set D, if $\lambda(D \backslash W) < q_D$ then $D \subseteq W$. The idea is to copy V_2. For each D, we calculate $\lambda(D \backslash W_2)$ sufficiently precisely so that if it is close to q_D (say significantly smaller that $2q_D$), then we add all of D to W.

We then let, for every clopen set D such that $D \nsubseteq W$ and $n < \omega$,

$$T_D(n) = \{k \,:\, B_{n,k} \cap D \subseteq W\}\,;$$

if $D \subseteq W$ then $T_D(n) = \varnothing$. We let

$$T(n) = \bigcup \{T_D(n) \,:\, D \text{ is a clopen set of granularity} \leqslant n\}\,.$$

We first observe that T is an h-trace, which amounts to showing, for all clopen D, that T_D is an h_D-trace. Suppose that $D \nsubseteq W$. For all $k \in T_D(n)$, $D \backslash W \subseteq B_{n,k}^{\complement}$; as the sets $B_{n,k}$ are independent,

$$q_D \leqslant \lambda(D \backslash W) \leqslant \left(1 - 2^{-n}\right)^{|T_D(n)|}\,,$$

so

$$|T_D(n)| \leqslant \log_{(1-2^{-n})}(q_D) = 1/\log_{q_D}(1 - 2^{-n}) \leqslant h_D(n).$$

Our task now is to show that T is computable, given W and $\lambda(W)$. To do this we show that using W and $\lambda(W)$, the sets $T_D(n)$ are computable, uniformly in D and n; and that we can, again uniformly in D and n, compute an upper bound for the elements of $T_D(n)$.

For the former, we observe that the collection of clopen subsets of W is computable from W and $\lambda(W)$. For $C \subseteq W$ if and only if $\lambda(C \backslash W) < q_C$ if and only if $\lambda(C \backslash W) = 0$, and $\lambda(C \backslash W)$ is computable, uniformly in C. So we can tell if $D \subseteq W$, and if not, then for all k, whether $B_{n,k} \cap D \subseteq W$ or not.

For the latter, suppose that D is clopen and $D \nsubseteq W$. So $\lambda(W|D) < 1$, where $\lambda(X|D)$ is the conditional probability. Given n, we find some large s so that

$$\lambda(W|D) - \lambda(W_s|D) < 2^{-n}(1 - \lambda(W|D)).$$

We also assume that s is greater than the granularity of D. We claim that $T_D(n) \subseteq s$. Suppose that $k \geqslant s$. Then $B_{n,k}$ and $W_s \cap D$ are independent, and also $B_{n,k}$ and D are independent; this implies that $B_{n,k}$ and W_s are independent modulo D, that is,

$$\lambda(W_s \cap B_{n,k}|D) = \lambda(W_s|D) \cdot \lambda(B_{n,k}|D) = 2^{-n}\lambda(W_s|D).$$

Then

$$\begin{aligned}\lambda(W_s \cup B_{n,k}|D) &= \lambda(W_s|D) + \lambda(B_{n,k}|D) - \lambda(W_s|D) \cdot \lambda(B_{n,k}|D) > \\ &\lambda(W|D) - 2^{-n}(1 - \lambda(W|D)) + 2^{-n} - 2^{-n}\lambda(W|D) \geqslant \lambda(W|D),\end{aligned}$$

so it is impossible that $B_{n,k} \cap D \subseteq W$. Hence $k \notin T_D(n)$.

Finally, we need to show that if $\bigcap U_n \subseteq \bigcap V_n$ then T traces f (where f is mapped to $\langle U_n \rangle$ and $\langle V_n \rangle$ is mapped to T). We show that there is some D such that T_D traces f.

We seek some clopen D such that $D \nsubseteq W$ but for all but finitely many m, $D \cap B_{m,f(m)} \subseteq W$; that is, $D \cap U_n \subseteq W$ for some n. If there is no such D then we build some $x \in (\bigcap_n U_n) \backslash W$ (but $\bigcap_n U_n \subseteq \bigcap_n V_n \subseteq V_2 \subseteq W$). This is done by initial segments; given σ_n such that $[\sigma_n] \nsubseteq W$, by assumption, $U_n \cap [\sigma_n] \nsubseteq W$, so we can find an extension $\sigma_{n+1} \succcurlyeq \sigma_n$ such that $[\sigma_{n+1}] \subseteq U_n$ but $[\sigma_{n+1}] \nsubseteq W$.[8] □

Corollary 3.15 (Bartoszynski [2]). *Let h be an order function.*

(a) $\mathbf{cof}(\mathscr{N})$ *is the smallest size of a family of h-traces that trace every function.*
(b) $\mathbf{add}(\mathscr{N})$ *is the smallest size of a family of functions in ω^ω which are not all traced by a single h-trace.*

Corollary 3.16 (Rupprecht [41, 42]). *An oracle is (strongly) null engulfing if and only if it is high.*

Corollary 3.17 (Terwijn and Zambella [45]). *An oracle is low for Schnorr tests if and only if it is computably traceable.*

Kjos-Hanssen, Nies and Stephan [28] showed that lowness for Schnorr randomness is also equivalent to being computably traceable. The framework discussed here does not appear to give tools for proving this equivalence. This is in contrast with lowness for genericity or Kurtz randomness, which we will discuss below.

3.6. **Reducing category to measure.** We prove:

Theorem 3.18. $\texttt{Supset}(\mathscr{M}) \to \texttt{Supset}(\mathscr{N})$.

We take a slightly roundabout way because below we will use the concepts that we introduce now. For a problem $A \subseteq A_{\texttt{inst}} \times A_{\texttt{sol}}$, we define a problem A^ω by letting $(A^\omega)_{\texttt{inst}} = (A_{\texttt{inst}})^\omega$, $(A^\omega)_{\texttt{sol}} = (A_{\texttt{sol}})^\omega$ and $(a_0, a_1, a_2, \dots)\ A^\omega\ (b_0, b_1, b_2, \dots)$ if and only if $a_0\, A\, b_0,\ a_1\, A\, b_1, \dots$. Note that we can think of $A^\omega_{\texttt{inst}}$ and $A^\omega_{\texttt{sol}}$ as subsets of ω^ω by the natural computable isomorphism between ω^ω and $(\omega^\omega)^\omega$.

It would be tempting to think that $A \to B$ implies $A^\omega \to B^\omega$; apply the maps $\psi_{\texttt{inst}}$ and $\psi_{\texttt{sol}}$ coordinate-wise. The problem is that these maps may fail to be hyperarithmetic piecewise computable; if $\psi_{\texttt{inst}} \upharpoonright A_n$ is computable for a partition $\langle A_n \rangle$, then $(\psi_{\texttt{inst}})^\omega$ is continuous on the sets $\prod_k A_{n_k}$, of which there are uncountably many. In the language of remark 2.4, if we need α jumps to determine which computable map to apply to x, then $\psi^\omega(x)$ will be $x^{(\alpha)}$-computable rather than x-computable.

However, if $\psi_{\texttt{inst}}$ and $\psi_{\texttt{sol}}$ are (total) computable functions, then this construction will work. In fact, slightly less is sufficient:

Lemma 3.19. Suppose that ψ is a morphism from A to B, that the sets $A_{\texttt{inst}}$ and $B_{\texttt{sol}}$ are hyperarithmetic, and that $\psi_{\texttt{inst}} \upharpoonright A_{\texttt{inst}}$ and $\psi_{\texttt{sol}} \upharpoonright B_{\texttt{sol}}$ are computable (i.e., restrictions of partial computable functions). Then $A^\omega \to B^\omega$.

Proof. $(A_{\texttt{inst}})^\omega$ and $(B_{\texttt{sol}})^\omega$ are hyperarithmetic, and the maps $(\psi_{\texttt{inst}})^\omega \upharpoonright (A_{\texttt{inst}})^\omega$ and $(\psi_{\texttt{sol}})^\omega \upharpoonright (B_{\texttt{sol}})^\omega$ are computable. Outside $(A_{\texttt{inst}})^\omega$ and $(B_{\texttt{sol}})^\omega$, we can use any constant computable function. □

Lemma 3.20. For all order functions h and $\tilde{h}$, $(h\text{-}\texttt{Trace})^\omega \leftrightarrow (\tilde{h}\text{-}\texttt{Trace})^\omega$.

[8] A quicker way to state this last argument is the following: the space $W^\complement = 2^\omega \backslash W$ is a Baire space. Restricted to this space, $\bigcap U_n = \varnothing$; so there is some n such that $U_n \backslash W$ is not dense in $W^\complement$. Thus there is some clopen D such that $D \cap W^\complement$ is nonempty but disjoint from U_n. The initial segment construction above reproves the Baire category theorem in the space $W^\complement$.

Proof. The maps in the morphism from h-`Trace` to $\tilde{h}$-`Trace` are computable. □

We therefore write $\texttt{Trace}^\omega$. However,

Lemma 3.21. $\texttt{Trace} \leftrightarrow \texttt{Trace}^\omega$.

Proof. For every problem A, $A \to A^\omega$: map $a \in A_{\texttt{inst}}$ to $(a, a, a \dots)$; on the solution side map $(b_0, b_1, \dots)$ to b_0.

Let $h(n,k) = \max\{n,k\}$. For this proof we choose a computable pairing function $(x,y) \mapsto \langle x,y\rangle$ which makes h non-decreasing (the value of the pair $\langle n,k\rangle$ is its location in a computable ordering of pairs which firsts orders by the maximum).

We reduce $(\mathrm{id}\text{-}\texttt{Trace})^\omega$ to h-`Trace`. On the instance side, map $\langle f_k\rangle$ to the function f defined by $f(k,n) = f_k(n)$. On the solution side, we map the h-trace T to the sequence of id-traces $(T_0, T_1, \dots)$, with $T_k(n) = \varnothing$ for $n < k$, and $T_k(n) = T(k,n)$ for $n \geqslant k$. □

Let $\mathscr{CND}$ denote the collection of closed, nowhere dense subsets of Cantor space, each named by a nowhere dense binary tree. $\texttt{Supset}(\mathscr{CND}, \mathscr{M})$ is the problem of finding a meagre superset of a given closed, nowhere dense set.

Proposition 3.22. $\texttt{Supset}(\mathscr{CND}, \mathscr{M}) \to \texttt{Trace}$.

To prove proposition 3.22 we need the following. Call a collection $\mathcal{O}$ of clopen subsets of Cantor space *dense* if for every dense open set $U \subseteq 2^\omega$ there is some $C \in \mathcal{O}$ such that $C \subseteq U$.

Lemma 3.23. There are uniformly computable dense families of clopen sets $\mathcal{O}_n$ such that for all n, the intersection of any n sets in $\mathcal{O}_n$ is nonempty.

Proof. For every pair $n < m < \omega$, let $A_{n,m}$ be the collection of functions g whose domain is 2^n, the set of binary strings of length n, such that for all $\sigma \in 2^n$, $g(\sigma)$ is a binary string of length m extending σ. Each such g represents a step toward defining a dense open set, by mapping strings of length n to extensions in the dense set we are building. Thus, for such g we let $C_g = \bigcup\{[\tau] : \tau \in \operatorname{range} g\}$. An open set $U \subseteq 2^\omega$ is dense if and only if for every n there is some $m > n$ and some $g \in A_{n,m}$ such that $C_g \subseteq U$.

We let $\mathcal{O}_n$ be the collection of unions $C_{g_1} \cup C_{g_2} \cup \dots \cup C_{g_n}$, where for some $0 = k_0 < k_1 < k_2 < \dots < k_n$ we have $g_i \in A_{k_{i-1},k_i}$. That is: the elements of $\mathcal{O}_n$ are obtained by choosing a string $[\sigma]$ of some length k_1, and enumerating it into the clopen set being build; then for each string of length k_1, we choose some extension of length k_2, and enumerating all such extensions; and so on, n times.

Suppose that $U \subseteq 2^\omega$ is dense and open. As described, for all n there is some $m > n$ and some $g \in A_{m,n}$ such that $C_g \subseteq U$. Thus, we can define by induction a sequence $k_1 < k_2 < \dots < k_n$ and find $g_i \in A_{k_{i-1},k_i}$ such that $C_{g_i} \subseteq U$. Thus, each $\mathcal{O}$ is a dense collection of clopen sets.

Let $n < \omega$ and let $D_1, D_2, \dots, D_n$ be elements of $\mathcal{O}_n$. For each $i \leqslant n$ there are $k^i_1 < k^i_2 < \dots < k^i_n$ and $g^i_j \in A_{k^i_{j-1},k^i_j}$ such that $D_i = \bigcup_{j \leqslant n} C_{g^i_j}$. We order these sets so that $k^1_1 \leqslant k^i_1$ for all $i \geqslant 1$; then $k^2_2 \leqslant k^i_2$ for all $i \geqslant 2$, and so on. Then $\bigcap_i C_{g^i_i} \subseteq \bigcap_i D_i$ and is nonempty: we define $\sigma_i \in \operatorname{range} g^i_i$ recursively, with $\sigma_1 \prec \sigma_2 \prec \dots \prec \sigma_n$, as $k^{i-1}_{i-1} \leqslant k^i_{i-1}$. □

Proof of proposition 3.22. $\texttt{Supset}(\mathscr{CND}, \mathscr{M})$ is of course the same as the problem of finding a dense Π^0_2 subset of a given dense open set; for this reduction it is easier to think about dense sets. Fix families $\mathcal{O}_n$ given by lemma 3.23. We order all finite binary strings in a computable list $\langle \sigma_n \rangle$. We reduce $\texttt{Supset}(\mathscr{CND}, \mathscr{M})$ to id-$\texttt{Trace}$.

On the instance side, we map a dense open set U to the function f defined as follows: $f(n)$ is some element C of $\mathcal{O}_n$ such that $\sigma_n \hat{} C = \{\sigma_n \hat{} x : x \in C\}$ is a subset of U. Such a clopen set exists because U is dense in $[\sigma_n]$.

On the solution side, let T be an id-bounded trace. We may assume that for all n, $T(n) \subseteq \mathcal{O}_n$. We map T to $\bigcap_k V_k$, where for $k < \omega$, V_k is the dense open set $\bigcup_{m>k} \sigma_m \hat{} \bigcap T(m)$. Since $|T(m)| \leqslant m$, $\bigcap T(m)$ is nonempty, and so V_n is dense open. If $g(m) \in T(m)$ for all $m > k$, then $V_k \subseteq U$, and so $\bigcap_k V_k \subseteq U$. □

The maps given in the proof above are computable on their arithmetic domains, and so by lemma 3.19, $(\texttt{Supset}(\mathscr{CND}, \mathscr{M}))^\omega \to (\texttt{Trace})^\omega$.

Proof of theorem 3.18. By theorem 3.14 and lemma 3.21, and the morphism $(\texttt{Supset}(\mathscr{CND}, \mathscr{M}))^\omega \to (\texttt{Trace})^\omega$, it suffices to show that $\texttt{Supset}(\mathscr{M})$ reduces to $(\texttt{Supset}(\mathscr{CND}, \mathscr{M}))^\omega$. But this is not difficult: on the instance side, a meagre set is given by a sequence $\langle T_n \rangle$ of nowhere dense sets, so we essentially take the identity map; on the solution side, we are given a sequence $\langle M_n \rangle$ of meagre sets; we can combine their presentations and map this sequence to $\bigcup_n M_n$. □

Corollary 3.24 (Bartoszynski [2]; Raisonnier, Stern [40])**.** $\mathbf{cof}(\mathscr{M}) \leqslant \mathbf{cof}(\mathscr{N})$ *and* $\mathbf{add}(\mathscr{N}) \leqslant \mathbf{add}(\mathscr{M})$.

Corollary 3.25. *Every computably traceable degree is low for meagre sets.*

Proof. By corollary 3.16; theorem 3.18 implies that every degree which is low for Schnorr tests is also low for meagre sets. □

For highness classes, the implication $\texttt{Supset}(\mathscr{M}) \to \texttt{Supset}(\mathscr{N})$ does not add computable information, as

$$\texttt{Dom} \to \texttt{Supset}(\mathscr{M}) \to \texttt{Supset}(\mathscr{N})$$

and we have already ascertained that $\mathrm{H}(\texttt{Supset}(\mathscr{N})) = \mathrm{H}(\texttt{Dom})$, as the former is $\mathrm{H}(\texttt{Trace})$.

3.7. **The Cichoń diagram.** The following diagram (fig. 2) displays all the morphisms for the problems associated with measure and category, as well as the domination problems. The analogous diagram for cardinal characteristics was named (by Fremlin) after Cichoń.

4. IOE reals and sequential composition

4.1. **Sequential composition.** A (total) function $F : \omega^\omega \to \omega^\omega$ is *Borel piecewise continuous* if there is a countable partition $\langle A_n \rangle$ of ω^ω into Borel sets such that for all n, $F \restriction A_n$ is continuous. We let $\mathcal{F}$ be the collection of these functions.

A *name* for such a function is an element of Baire space coding:

- The sets A_n: for each n, some oracle z_n and a $\Delta^1_1(z_n)$-definition of A_n;
- The functions $F \restriction A_n$: a name for a partial continuous function ψ_n which agrees with F on A_n.

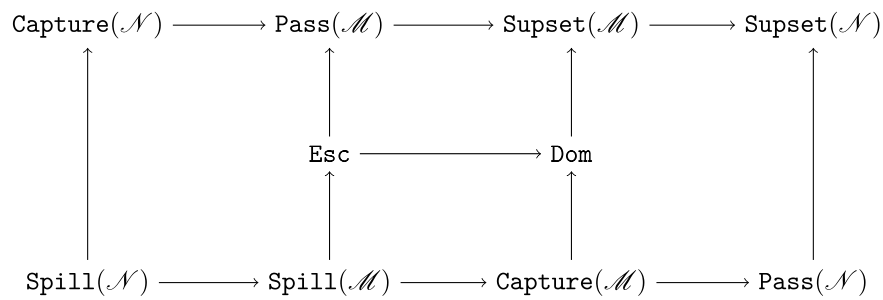

FIGURE 2. The Cichoń diagram for Weihrauch problems

Note that the collection of names is quite complicated (d-Π^1_1). For $x \in 2^\omega$, a Borel piecewise continuous function is *x-computable* if it has an x-computable name. If $F \in \mathcal{F}$ is x-computable then for all $a \in \omega^\omega$, $F(a) \leqslant_{\mathrm{T}} (x, a)$. The naming system is chosen so that the Borel piecewise continuous maps with computable names are precisely the hyperarithmetic piecewise computable ones.

Lemma 4.1. $\mathcal{F}$ has effective composition: there is a computable map $\theta\colon (\omega^\omega)^2 \to \omega^\omega$ such that if f, g are names for $F, G \in \mathcal{F}$ then $\theta(f, g)$ is a name for $F \circ G$.

Proof. Let $\langle A_n \rangle, \langle B_n \rangle$ be the partitions. We let $C_{n,m} = A_n \cap F^{-1}[B_m] = (F \upharpoonright A_n)^{-1}[B_m]$; otherwise, we compose partial continuous functions. □

Every constant function is in $\mathcal{F}$; the constant function $y \mapsto a$ has an a-computable name.

For problems A and B we define the problem $A \star B$. The instances are pairs $(a, F) \in A_{\mathtt{inst}} \times \mathcal{F}$ (as usual we mean a name for $F \in \mathcal{F}$) such that $F[A_{\mathtt{sol}}] \subseteq B_{\mathtt{inst}}$; the solutions are pairs in $A_{\mathtt{sol}} \times B_{\mathtt{sol}}$. The relation is: (c, d) is an $(A \star B)$-solution of (a, F) if c is an A-solution for a and d is a B-solution for $F(c)$. The idea is to take a two-step iteration: first solve an A-instance a; use the solution to generate a new B-instance, and then solve that one.[9]

The collection of instances of $A \star B$ can be quite complicated. However it does contain computable points: use a constant function that maps to a computable instance of B.

We now verify that sequential operation induces a well-defined operation on the morphism equivalence classes of Weihrauch problems.

Lemma 4.2. $A, B \to A \star B$

Proof. To reduce A to $A \star B$ we use the idea just mentioned: fix a computable $b_* \in B_{\mathtt{inst}}$; let F_{b*} be the constant function $y \mapsto b_*$. We map an A-instance a to (a, F_{b*}) and map an $(A \star B)$-solution (c, d) to c.

To reduce B to $A \star B$, fix a computable A-instance a_*. We map a B-instance b to the pair (a_*, F_b) (where again F_b is the constant function $y \mapsto b$); we map an $(A \star B)$-solution (c, d) to d. □

[9]Blass [7] defines $A \star B$ using the collection of all functions from $A_{\mathtt{sol}}$ to $B_{\mathtt{inst}}$. As this collection cannot be represented, Rupprecht did not try to find a computable analogue. A similar yet non-identical sequential composition was discovered by Brattka et al. [9], who only used computable maps. Our formulation, and choice of functions for morphisms, are designed to obtain lemma 4.3.

Lemma 4.3. If $A \to \hat{A}$ and $B \to \hat{B}$ then $(A \star B) \to (\hat{A} \star \hat{B})$.

Proof. Let ψ be a morphism from A to $\hat{A}$ and let φ be a morphism from B to $\hat{B}$. Let (a, F) be an $(A \star B)$-instance. Let $\hat{a} = \psi_{\mathtt{inst}}(a)$ and let $\hat{F}$ be the translation of F under the relevant morphisms: $\hat{F} = \varphi_{\mathtt{inst}} \circ F \circ \psi_{\mathtt{sol}}$. As $F\colon A_{\mathtt{sol}} \to B_{\mathtt{inst}}$, it follows that $\hat{F}\colon \hat{A}_{\mathtt{sol}} \to \hat{B}_{\mathtt{inst}}$. We send (a, F) to $(\hat{a}, \hat{F})$, using lemma 4.1 to see that this map is hyperarithmetic piecewise computable. Let $(\hat{c}, \hat{d})$ be an $\hat{A} \star \hat{B}$-solution. We map it to $(c, d) = (\psi_{\mathtt{sol}}(\hat{c}), \varphi_{\mathtt{sol}}(\hat{d}))$. □

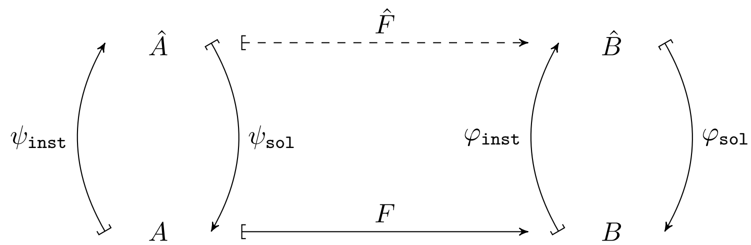

FIGURE 3. Reducing $A \star B$ to $\hat{A} \star \hat{B}$.

Lemma 4.4. $(A \star B) \star C \leftrightarrow A \star (B \star C)$.

Proof. This goes as expected. Abusing notation, we write X^Y for the collection of maps from Y to X that are in $\mathcal{F}$. So $(A \star B)_{\mathtt{inst}} = A_{\mathtt{inst}} \times (B_{\mathtt{inst}})^{A_{\mathtt{sol}}}$. The proof of the lemma relies on the natural identification of

$$((A \star B) \star C)_{\mathtt{inst}} = A_{\mathtt{inst}} \times (B_{\mathtt{inst}})^{A_{\mathtt{sol}}} \times (C_{\mathtt{inst}})^{A_{\mathtt{sol}} \times B_{\mathtt{sol}}}$$

with

$$(A \star (B \star C))_{\mathtt{inst}} = A_{\mathtt{inst}} \times \left(B_{\mathtt{inst}} \times (C_{\mathtt{inst}})^{B_{\mathtt{sol}}}\right)^{A_{\mathtt{sol}}}.$$ □

When describing a map $\varphi_{\mathtt{inst}} : C_{\mathtt{inst}} \to (A \star B)_{\mathtt{inst}}$, we must send $c \in C_{\mathtt{inst}}$ to some $(a, F) \in (A \star B)_{\mathtt{inst}}$, where a Borel piecewise continuous name for F is obtained in some hyperarithmetic piecewise computable fashion from a. Our description of F can thus make use of a. If the Weihrauch problem A is hyperarithmetic, that is, if $\{y \in A_{\mathtt{sol}} : xAy\}$ is uniformly hyperarithmetic in x, then it suffices to describe $F(y)$ under the assumption that aAy; we can then define F to some computable value for all other y while still obtaining an a-computable Borel piecewise continuous name.

Proposition 4.5.

(a) *If* $\mathrm{Card}(A)$ *and* $\mathrm{Card}(B)$ *are infinite, then* $\mathrm{Card}(A \star B) = \max\{\mathrm{Card}(A), \mathrm{Card}(B)\}$.

(b) $\mathrm{NL}(A \star B) = \mathrm{NL}(A) \cup \mathrm{NL}(B)$.

Proof. (a): Let Z be a complete solution set for A and W be a complete solution set for B. Then $Z \times W$ is a complete solution set for $A \star B$. On the other hand, since $A, B \to A \star B$, $\max\{\mathrm{Card}(A), \mathrm{Card}(B)\} \leqslant \mathrm{Card}(A \star B)$.

(b): Again by lemma 4.2, $\mathrm{NL}(A), \mathrm{NL}(B) \subseteq \mathrm{NL}(A \star B)$. Suppose that $x \in \mathrm{NL}(A \star B)$. Let (a, F) be an x-computable $(A \star B)$-problem with no computable solution. If a has no computable A-solution then we are done. Otherwise let c be a computable A-solution for a. Let $b = F(c)$. Since F is x-computable, so is b. And it has no computable B-solution. □

Proposition 4.6. $\mathrm{Card}((A \star B)^\perp) = \min\{\mathrm{Card}(A^\perp), \mathrm{Card}(B^\perp)\}$.

Proof. We know that $(A \star B)^\perp \to A^\perp, B^\perp$. So it remains to show that

$$\mathrm{Card}((A \star B)^\perp) \geqslant \min\{\mathrm{Card}(A^\perp), \mathrm{Card}(B^\perp)\}.$$

Let R be a complete set of solutions for $(A \star B)^\perp$ of size $\mathrm{Card}((A \star B)^\perp)$. Let $P = \pi_0[R] = \{c : (\exists G)\,(c, G) \in R\}$. Then $|P| \leqslant |R|$. If P is a complete solution set for $A^\perp$ then $\mathrm{Card}(A^\perp) \leqslant \mathrm{Card}((A \star B)^\perp)$. Suppose not. Then there is some $a^* \in A_{\mathtt{sol}}$ such that for all $c \in P$, cAa^*. We let $Q = \{G(a^*) : (\exists c)\,(c, G) \in R\}$. Then $|Q| \leqslant |R|$. We claim that Q is a complete solution set for $B^\perp$, so $\mathrm{Card}(B^\perp) \leqslant \mathrm{Card}((A \star B)^\perp)$. For suppose that $b \in B_{\mathtt{sol}}$. Then (a^*, b) is an instance of $(A \star B)^\perp$, so there is some $(c, G) \in R$ which solves it. This means that either $c\not{A}a^*$, or that $G(a^*)\not{B}b$, that is, $bB^\perp G(a^*)$. The former is not true, so $G(a^*) \in Q$ is a $B^\perp$-solution of b. ☐

For the following lemma, we generalise notation: for $x \in 2^\omega$ and a problem A, $\mathrm{H}^x(A)$ is the set of $y \in 2^\omega$ which compute $c \in A_{\mathtt{sol}}$ which solves every x-computable A-instance.

Lemma 4.7. $y \in \mathrm{H}(A \star B)$ if and only if $y \in \mathrm{H}^x(B)$ for some $x \in \mathrm{H}(A)$ such that $x \leqslant_{\mathrm{T}} y$.

Proof. The point is that for all $z, w \in \omega^\omega$, $z \leqslant_{\mathrm{T}} w$ if and only if there is some computable $F \in \mathcal{F}$ such that $F(w) = z$. This is because every partial computable function can be extended to a computable $F \in \mathcal{F}$. ☐

4.2. **A weakening of morphism reduction.** For a Weihrauch problem A and $n < \omega$, let

$$A^{\star n} = \underbrace{A \star A \star \cdots \star A}_{n \text{ times}}$$

(we implicitly use the associativity of sequential composition, lemma 4.4). For Weihrauch problems A and B we write

$$A \dashrightarrow B$$

if $A \to B^{\star n}$ for some $n < \omega$. We also write $A \dashleftarrow\!\!\dashrightarrow B$ if $A \dashrightarrow B$ and $B \dashrightarrow A$. Lemma 4.2 and propositions 4.5 and 4.6 imply:

Lemma 4.8.

(a) $\dashrightarrow$ is transitive, and $\dashleftarrow\!\!\dashrightarrow$ is an equivalence relation.
(b) If $A \dashrightarrow B$ then $\mathrm{Card}(A) \leqslant \mathrm{Card}(B)$ and $\mathrm{Card}(A^\perp) \geqslant \mathrm{Card}(B^\perp)$.
(c) If $A \dashrightarrow B$ then $\mathrm{NL}(A) \subseteq \mathrm{NL}(B)$.

4.3. **An example: $\mathtt{Supset}(\mathscr{M})$.**

Proposition 4.9. $\mathtt{Supset}(\mathscr{M}) \to \mathtt{Pass}(\mathscr{M}) \star \mathtt{Dom}$.

Proof. The first step is to replace $\mathtt{Pass}(\mathscr{M})$ with a fractal version: $\mathtt{FractalPass}(\mathscr{M})$ is the problem of finding, given a meagre set A, a point $z \in \omega^\omega$ such that for all $w =^* z$, $w \notin A$. Clearly $\mathtt{Pass}(\mathscr{M}) \to \mathtt{FractalPass}(\mathscr{M})$. To show the reverse morphism, we map the meagre set A (an instance of $\mathtt{FractalPass}(\mathscr{M})$) to $\bigcup_{\sigma \in \omega^{<\omega}} (\sigma^\frown A)$. Here, for $x \in \omega^\omega$, $\sigma^\frown x$ is the result of replacing the first $|\sigma|$ values of x by σ, and $\sigma^\frown X = \{\sigma^\frown x : x \in X\}$. On the solution side, we map a real to itself.

Now we show $\texttt{Supset}(\mathscr{M}) \to \texttt{FractalPass}(\mathscr{M}) \star \texttt{Dom}$. On the instance side, we map $A = \bigcup_n [T_n]$ to the pair (A, F), where $F(z)$ is defined as follows. Suppose that $w \notin A$ for all $w =^* z$. Then $F(z)$ is the function f mapping (τ, n) to $k \geqslant |\tau|$ such that $(\tau^\frown z)\upharpoonright k \notin T_m$ for all $m \leqslant n$.

On the solution side, we map (z, g) to the meagre set $\bigcup B_n$, where for each τ, $[(\tau^\frown z)\upharpoonright g(\tau, n)]$ is thrown out of B_n. If $g(-, n)$ majorises $f(-, n)$ then $[T_m] \subseteq B_n$ for all $m \leqslant n$. □

We already know that $\texttt{Pass}(\mathscr{M}), \texttt{Dom} \to \texttt{Supset}(\mathscr{M})$. It follows that

$$\texttt{Pass}(\mathscr{M}) \star \texttt{Dom} \to \texttt{Supset}(\mathscr{M}) \star \texttt{Supset}(\mathscr{M}),$$

whence $\texttt{Pass}(\mathscr{M}) \star \texttt{Dom} \dashrightarrow \texttt{Supset}(\mathscr{M})$, which with proposition 4.9 gives

$$\texttt{Supset}(\mathscr{M}) \dashleftarrow\dashrightarrow \texttt{Pass}(\mathscr{M}) \star \texttt{Dom}.$$

As a result:

Corollary 4.10.

(a) $\mathbf{cof}(\mathscr{M}) = \max\{\mathfrak{d}, \mathbf{non}(\mathscr{M})\}$ *(A. Miller and Truss);*
(b) $\mathbf{add}(\mathscr{M}) = \min\{\mathfrak{b}, \mathbf{cov}(\mathscr{M})\}$ *(Fremlin);*
(c) *A degree is not low for meagre sets if and only if it is either hyperimmune or weakly meagre engulfing.*

Examining the morphisms, we also get that if a degree $\mathbf{d}$ is high relative to a weakly 1-generic below it then it is strongly meagre engulfing. However we already know that highness suffices, indeed is equivalent.

4.4. **Infinitely often equal reals.** $\texttt{AEDiff}$ is the problem of finding a function g which is different from a given f on all but finitely many inputs. Its dual is $\texttt{IOE}$, the problem of finding a function f which agrees with a given g on infinitely many inputs.

$\mathrm{H}(\texttt{IOE}) = \mathrm{NL}(\texttt{AEDiff})$ is the class of degrees computing functions f, which agree with every computable function on infinitely many inputs. Similarly, the dual class $\mathrm{NL}(\texttt{IOE}) = \mathrm{H}(\texttt{AEDiff})$ is that of degrees computing functions which are different from every computable function on all but finitely many inputs.

Kjos-Hanssen, Merkle and Stephan [27] showed:

Proposition 4.11. $\mathrm{NL}(\texttt{IOE})$ *is the class of degrees which are either high or* DNR*.*

We consider other known morphisms and equivalences.

Proposition 4.12. $\texttt{IOE} \to \texttt{Pass}(\mathscr{M})$*.*

Proof. We prove the equivalent $\texttt{Capture}(\mathscr{M}) \to \texttt{AEDiff}$. On the instance side, we start with a point $f \in \omega^\omega$; we map it to itself. On the solution side, we map $g \in \omega^\omega$ to the meagre set of $h \in \omega^\omega$ which are almost always different from g. □

Proposition 4.13. $\texttt{AEDiff} \to \texttt{Dom}$*.*

Proof. If $f \leqslant^* g$ then $g + 1$ is almost always different from f. □

Note how the morphism from $\texttt{Capture}(\mathscr{M})$ to $\texttt{Dom}$ (proposition 3.3) is close to the composition of the morphisms from $\texttt{Capture}(\mathscr{M})$ to $\texttt{AEDiff}$ and $\texttt{AEDiff}$ to $\texttt{Dom}$ given by propositions 4.12 and 4.13.

Proposition 4.14. $\texttt{Pass}(\mathscr{M}) \to \texttt{IOE} \star \texttt{IOE}$*.*

Proof. We prove that $\mathtt{Pass}(\mathscr{M}) \to \mathtt{Esc} \star \mathtt{IOE}$; since $\mathtt{Esc} \to \mathtt{IOE}$ (the dual of proposition 4.13), lemma 4.3 says that $\mathtt{Esc} \star \mathtt{IOE} \to \mathtt{IOE} \star \mathtt{IOE}$.

We work in Cantor space. On the instance side, we are given a meagre set A with name $\langle T_n \rangle$. As usual we may assume that $T_n \subseteq T_{n+1}$. In this proof, a *partial string* is a finite function to $\{0,1\}$. We say that a partial string τ *witnesses the meagreness of* A if the domain of τ is an interval $[n,m)$ and for every $\sigma \in 2^{<\omega}$ of length n, $\sigma \cup \tau \notin T_n$. Effectively from $\langle T_n \rangle$ we obtain a function $h\colon \omega \to \omega$ such that for all n, there is some partial string $\tau\colon [n, h(n)) \to \{0,1\}$ which witnesses the meagreness of A. Note that if $x \in 2^\omega$ and for infinitely many n there is some $m \geqslant n$ such that $x\restriction[n,m)$ witnesses the meagreness of A, then $x \notin A$.

Given a function $g \in \omega^\omega$, consider the sequence of intervals $\langle I_n \rangle$ given by $I_n = [g^{(n)}(0), g^{(n+2)}(0))$; note that this is not a partition of ω: these intervals overlap. If g escapes h then there are infinitely many n for which there are partial strings τ such that $\operatorname{dom}\tau \subseteq I_n$ and τ witnesses the meagreness of A.

Let σ, τ be two partial strings. We say that σ and τ are *strongly compatible* (relative to g) if for some n and m, $\operatorname{dom}\sigma \subseteq I_n$, $\operatorname{dom}\tau \subseteq I_m$ and the intervals I_n and I_m are disjoint. We now define $F(g)$: assuming g escapes h, $F(g)$ is a function f such that for all k, $f(k)$ is a set of $3k+1$ many finite functions τ, pairwise strongly compatible (relative to g), each of which witnesses the meagreness of A, such that $\min\operatorname{dom}\tau \geqslant k$. On the instance side, we map A to the pair (h, F).

On the solution side, we are given a pair (g,p) of functions. Given g we can compute $\langle I_n \rangle$, and so the notion of strong compatibility of partial strings relative to g.[10] We define a sequence $\langle \sigma_k \rangle$ of pairwise strongly compatible partial strings. Given $\sigma_0, \dots, \sigma_{k+1}$, we examine $p(k)$. We may assume that this is a set of $3k+1$ many pairwise strongly compatible partial strings τ such that $\min\operatorname{dom}\tau \geqslant k$ (if not, we pick σ_k to be empty). One of these partial strings is strongly compatible with each σ_i for $i < k$, and so can be chosen as σ_k. We then map the pair (g,p) to $x \in 2^\omega$ defined by extending $\bigcup_k \sigma_k$; we fill the undetermined locations arbitrarily. The condition $\min\operatorname{dom}\sigma_k \geqslant k$ implies that x can be computed from (g,p), uniformly.

If indeed A is mapped to (f,H) and (g,p) solves (f,H), then g escapes f and for infinitely many k, $p(k) = F(g)(k)$, in which case σ_k witnesses the meagreness of A. It then follows that $x \notin A$, as required. □

Propositions 4.12 and 4.14 give:

$$\mathtt{IOE} \longleftrightarrow \mathtt{Pass}(\mathscr{M}),$$

which yields some characterisations of cardinals and non-lowness classes, due to Rupprecht [41] and Bartoszynski [3].

Corollary 4.15.

(a) *A degree is weakly meagre engulfing if and only if it is high or* DNR*.*
(b) *A degree is not low for meagre sets if and only if it is hyperimmune or* DNR*.*
(c) $\mathrm{Card}(\mathtt{IOE}) = \mathbf{non}(\mathscr{M})$ *and* $\mathrm{Card}(\mathtt{AEDiff}) = \mathbf{cov}(\mathscr{M})$*.*

Proof. (a) follows from $\mathrm{NL}(\mathtt{IOE}) = \mathrm{NL}(\mathtt{Pass}(\mathscr{M}))$, and proposition 4.11.

(b) follows from (a) and corollary 4.10(b). □

[10] Of course, in the absence of A, we do not know which partial strings witness the meagreness of A.

As for NL($\texttt{AEDiff}$), the string of morphisms

$$\texttt{Capture}(\mathscr{M}) \to \texttt{AEDiff} \to \texttt{Dom}$$

and the equivalence of hyperimmune with computing a weakly 1-generic sandwich the class in the middle, yielding NL($\texttt{AEDiff}$) being hyperimmune as well.

4.5. **Lowness for closed nowhere dense sets, and weak 1-genericity.** Corollary 4.15(b) is very close to a result of Stephan and Yu's [44]. Rather than lowness for meagre sets, they consider the related notion of lowness for closed nowhere dense sets. That is, NL($\texttt{Supset}(\mathscr{CND})$) rather than NL($\texttt{Supset}(\mathscr{M})$). They show that this class is also equivalent to being hyperimmune or DNR; that is, NL($\texttt{Supset}(\mathscr{CND})$) = NL($\texttt{Supset}(\mathscr{M})$). They further show that this class coincides with non-lowness for weak 1-genericity, a property which doesn't seem to be expressed as the highness class of a Weihrauch problem.

We note that we cannot get a morphism equivalence between $\texttt{Supset}(\mathscr{M})$ and $\texttt{Supset}(\mathscr{CND})$; the corresponding non-lowness classes coincide, but their highness classes do not: indeed, H($\texttt{Supset}(\mathscr{CND})$) is empty, as the union of all computable closed, nowhere dense sets is dense. Nonetheless, we have the means to deduce the Stephan-Yu results. We now work in Cantor space, though the results can be extended to Baire space as well.

In one direction we do get a morphism:

Proposition 4.16. *$\texttt{Supset}(\mathscr{M}) \to \texttt{Supset}(\mathscr{CND})$.*

Proof. Recall the problems A^ω mentioned above. First, we show that $\texttt{Supset}(\mathscr{M}) \to (\texttt{Supset}(\mathscr{CND}))^\omega$. This is not hard: on both sides, map a sequence $\langle T_n \rangle$ to itself; we just consider them differently, as a name for a meagre set or an ω-sequence of names for closed, nowhere dense sets. If $\langle S_n \rangle$ is a $(\texttt{Supset}(\mathscr{CND}))^\omega$-solution for $\langle T_n \rangle$, that is, if $[T_n] \subseteq [S_n]$ for all n, then certainly $\bigcup_n [T_n] \subseteq \bigcup_n [S_n]$.

Next, we show that $\texttt{Supset}(\mathscr{CND}) \leftrightarrow (\texttt{Supset}(\mathscr{CND}))^\omega$. As observed above, we always have $A \mapsto A^\omega$. We reduce $(\texttt{Supset}(\mathscr{CND}))^\omega$ to $\texttt{Supset}(\mathscr{CND})$. On the instance side, map $\langle T_n \rangle$ to the tree $S = \bigcup_n (\{0^n\} \cup 0^n 1\hat{\ } T_n)$. That is, paste T_n on the n^{th} level of a fishbone. On the solution side undo this operation. ☐

We now obtain:

Proposition 4.17. *NL($\texttt{Supset}(\mathscr{M})$) = NL($\texttt{Supset}(\mathscr{CND})$).*

Proof. Proposition 4.16 shows that NL($\texttt{Supset}(\mathscr{M})$) ⊆ NL($\texttt{Supset}(\mathscr{CND})$). In the other direction, the proof of proposition 4.9 shows that $\texttt{Supset}(\mathscr{CND}) \to \texttt{Pass}(\mathscr{M}) \star \texttt{Maj}$, where $\texttt{Maj}$ is the problem of finding a function g that majorises a given function f, that is, $(\forall n)\, f(n) \leqslant g(n)$. While H($\texttt{Maj}$) is empty, NL($\texttt{Maj}$) = NL($\texttt{Dom}$) is the collection of hyperimmune degrees. It follows that

$$\text{NL}(\texttt{Supset}(\mathscr{CND})) \subseteq \text{NL}(\texttt{Pass}(\mathscr{M})) \cup \text{NL}(\texttt{Maj}) = \text{NL}(\texttt{Supset}(\mathscr{M})).$$ ☐

What about lowness for weak genericity? Here we can use a technique utilised by Greenberg and Monin [20]. We dualise and define the Weihrauch problem:

- $\texttt{FractalSpill}(\mathscr{CND})$: an instance is a closed, nowhere dense set P; a solution is a closed, nonempty, nowhere dense set Q and an infinite set $I \subseteq \omega$ such that $0 \in I$ and for all $n \in I$, for all $\sigma \in 2^{<\omega}$ of length n such that $[\sigma] \cap Q \neq \varnothing$, $Q \cap [\sigma] \nsubseteq \sigma\hat{\ }P$. (Recall that $\sigma\hat{\ }P = \{\sigma\hat{\ }x : x \in P\}$.)

Lemma 4.18. $\texttt{Spill}(\mathscr{CND}) \leftrightarrow \texttt{FractalSpill}(\mathscr{CND})$.

Proof. $\texttt{Spill}(\mathscr{CND}) \to \texttt{FractalSpill}(\mathscr{CND})$ is immediate, via identity maps, forgetting the set I. For the other direction, we use the construction from [20]. On the instance side we use the identity map. On the solution side, given a nowhere dense tree T, we define a nowhere dense tree S and an infinite set I. We start with $0 \in I$ and $\langle\rangle \in S$. Suppose that we have declared that $n \in I$ and for every string ρ of length at most n, whether $\rho \in S$ or not. For each string $\sigma \in S$ of length n, the tree $\bigcup_{k \leqslant |\sigma|} (\sigma \upharpoonright k)\hat{}T$ is nowhere dense, so we can find some $\tau \succcurlyeq \sigma$ off that tree. We declare that $\tau \notin S$. We let the next element of I be bigger than all these τ's.

By construction, for every string $\sigma \in S$ such that $|\sigma| \in I$, $\sigma\hat{}T \subseteq S$. Suppose that $|\sigma| \in I$, $\sigma \in S$ and $[\sigma] \cap [S] \subseteq \sigma\hat{}P$. Then $\sigma\hat{}[T] \subseteq \sigma\hat{}P$, whence $[T] \subseteq P$. ☐

Lemma 4.19. Suppose that Γ is a countable collection of closed, nowhere dense sets, which is closed under the shift operator: for all σ and $P \in \Gamma$, $P - \sigma = \{x : \sigma\hat{}x \in P\}$ is also in Γ. Suppose that (Q, I) is a $\texttt{FractalSpill}(\mathscr{CND})$-solution for every $P \in \Gamma$. Then $Q \nsubseteq \bigcup \Gamma$. Thus, an oracle computing such Q is not low for Γ-genericity.

Proof. The cloure property of Γ means that for every $P \in \Gamma$ and every σ such that $|\sigma| \in I$ and $[\sigma] \cap Q \neq \varnothing$, $[\sigma] \cap Q \nsubseteq P$. This is because $P - \sigma \in \Gamma$ and $[\sigma] \cap Q \nsubseteq \sigma\hat{}(P - \sigma) = P \cap [\sigma]$.

We can therefore build a point in $Q \backslash \bigcup \Gamma$ by finite extension. At stage k we have a string σ_k such that $[\sigma_k] \cap Q \neq \varnothing$ and $|\sigma_k| \in I$. Let P_k be the k^{th} element of Γ in some ω-enumeration of Γ. Since $[\sigma_k] \cap Q \nsubseteq P_k$, we can find an extension σ_{k+1} of σ_k such that $[\sigma_{k+1}] \cap Q \neq \varnothing$, $|\sigma_{k+1}| \in I$ and $[\sigma_{k+1}] \cap P_k = \varnothing$. ☐

We now get the full result of Stephan and Yu, which is also another proof of proposition 4.17.

Corollary 4.20.

$\mathrm{NL}(\texttt{Supset}(\mathscr{M})) = \mathrm{NL}(\texttt{Supset}(\mathscr{CND})) = \{x : x \textit{ is not low for weak 1-genericity}\}$.

Proof. By proposition 4.16, $\mathrm{NL}(\texttt{Supset}(\mathscr{M})) \subseteq \mathrm{NL}(\texttt{Supset}(\mathscr{CND}))$. By lemma 4.18 and lemma 4.19, applied to $\Gamma = \Delta^0_1 \cap \mathscr{CND}$, we see that every oracle in $\mathrm{NL}(\texttt{Supset}(\mathscr{CND}))$ is not low for weak 1-genericity. Finally, suppose that x is not low for weak 1-genericity; say g is weakly 1-generic, $M \in \mathscr{M}$ is x-computable and $g \in M$; then M shows that $x \in \mathrm{NL}(\texttt{Supset}(\mathscr{M}))$. ☐

4.6. **A non-morphism.** The implications

$$\texttt{Supset}(\mathscr{M}) \to \texttt{Pass}(\mathscr{M}) \star \texttt{Dom}$$

and

$$\texttt{IOE} \to \texttt{Pass}(\mathscr{M}) \to \texttt{IOE} \star \texttt{IOE}$$

leave the question open whether we can get morphism reverals, that is, if we really needed the weakening of morphism equivalence. For computable morphisms, we can use highness classes.

Lemma 4.21. $\texttt{Pass}(\mathscr{M}) \star \texttt{Dom} \nrightarrow \texttt{Supset}(\mathscr{M})$.

Proof. There is a high degree which is not high relative to a weakly 1-generic below it, e.g. a minimal high degree. ☐

Lemma 4.22. $\texttt{IOE} \star \texttt{IOE} \nrightarrow \texttt{IOE}$.

Proof. We show that $\mathrm{H}(\texttt{IOE}) \neq \mathrm{H}(\texttt{IOE} \star \texttt{IOE})$. By lemma 4.7, $y \in \mathrm{H}(\texttt{IOE} \star \texttt{IOE})$ if and only if there is some $x \leqslant_{\mathrm{T}} y$ in $\mathrm{H}(\texttt{IOE})$ such that $y \in \mathrm{H}^x(\texttt{IOE})$. We observed that $\mathrm{H}(\texttt{IOE}) = \mathrm{NL}(\texttt{AEDiff})$ is the collection of oracles of hyperimmune degree. Relativising, $y \in \mathrm{H}(\texttt{IOE} \star \texttt{IOE})$ if and only if y has hyperimmune degree relative to a hyperimmune $x \leqslant_{\mathrm{T}} y$. We cannot have $x \equiv_{\mathrm{T}} y$, in particular, y cannot have minimal degree. On the other hand, there is a hyperimmune x of minimal degree. □

These fine distinctions do not help if we change the question to the existence of a definable (say, Borel) morphism. We shall get back to this question in section 8.

5. Other reducibilities

We can modify the definition of the highness class (and thus the non-lowness class) associated with a Weihrauch problem. The most obvious one is changing the reducibility from Turing to weaker ones. If $\leqslant_r$ is any reducibility, implied by Turing, then for a Weihrauch problem A we define $\mathrm{H}^r(A)$ to be the collection of oracles x such that there is some $c \in A_{\texttt{sol}}$, $c \leqslant_r x$ which is an A-solution for every $a \in A_{\texttt{inst}}$ such that $a \leqslant_r \varnothing$.

Morphisms give implications for these variants of highness and non-lowness classes.

Lemma 5.1. Suppose that $\leqslant_r$ is a transitive relation implied by Turing reducibility. Let A and B be Weihrauch problems. Then:

(a) If $A \to B$ then $\mathrm{NL}^r(A) \subseteq \mathrm{NL}^r(B)$.
(b) $\mathrm{NL}^r(A \star B) = \mathrm{NL}^r(A) \cup \mathrm{NL}^r(B)$.

As a result, if $A \dashrightarrow B$ then $\mathrm{NL}^r(A) \subseteq \mathrm{NL}^r(B)$.

The most commonly used in this area is hyperarithmetic reducibility, equivalent to relatively Δ^1_1 reducibility. Thus $\mathrm{NL}^{\Delta^1_1}(A)$ is the collection of $x \in 2^\omega$ such that there is some $c \in A_{\texttt{inst}} \cap \Delta^1_1(x)$ which has no Δ^1_1 A-solution. The relationships between problems studied above give analogous results in this context:

Theorem 5.2 (Chong, Nies, Yu [12])**.** *An oracle is low for Δ^1_1 null sets if and only if it is Δ^1_1-traceable.*

Theorem 5.3 (Greenberg, Monin [20])**.** *An oracle x is low for Δ^1_1 genericity if and only if it is low for Δ^1_1 closed nowhere dense sets if and only if it is Δ^1_1-dominated and every $f \in \Delta^1_1(x)$ is infinitely often equal to some Δ^1_1 function.*

We also obtain results that we believe have not been stated yet, for example:

Proposition 5.4. *There is a meagre set in $\Delta^1_1(x)$ containing all Δ^1_1 points if and only if there is a function $f \in \Delta^1_1(x)$ which is eventually different from every Δ^1_1 function.*

In section 8 below we discuss another family of weak reducibilities $\leqslant_r$, namely Turing modulo an ideal.

5.1. Total reductions.

In the other direction, we can ask what happens when we strengthen, rather than weaken, Turing reducibility. This has been investigated, for example, by Miyabe [32]. We make the following definition.

Definition 5.5. Let $X \subseteq \omega^\omega$. We say that $a \in X$ is X-$\mathtt{tt}$ *reducible* to $x \in 2^\omega$ if there is a (total) computable function $I\colon 2^\omega \to X$ such that $a = I(x)$. We write $a \leqslant_{X\text{-}\mathtt{tt}} x$.

Let A be a Weihrauch problem. We let $\mathrm{NL}^{\mathtt{tt}}(A)$ be the collection of oracles $x \in 2^\omega$ such that there is some $a \leqslant_{A_{\mathtt{inst}}\text{-}\mathtt{tt}} x$ which has no computable A-solution.

Sometimes this notion trivialises, for example $\mathrm{NL}^{\mathtt{tt}}(\mathtt{Dom}) = \varnothing$. In other cases, we get something meaningful. Miyabe, in particular, examines what he calles *uniform* Schnorr randomness, where a uniform oracle test is an operator which gives a Schnorr test relative to every oracle. That is, a uniform Schnorr test relative to x is an element $N \in \mathscr{N}$ which is $\mathscr{N}$-$\mathtt{tt}$-reducible to x.

To use reducibilities in this context, we need to maintain totality.

Lemma 5.6. Suppose that $A \to B$ by a morphism ψ such that $\psi_{\mathtt{inst}} \upharpoonright A_{\mathtt{inst}}$ is computable. As usual, this means it is the restriction to $A_{\mathtt{inst}}$ of a partial computable function. Then $\mathrm{NL}^{\mathtt{tt}}(A) \subseteq \mathrm{NL}^{\mathtt{tt}}(B)$.

We observe that all the morphisms we have considered so far except for one (remark 3.8) are uniform on their instances. We thus get:

Theorem 5.7 (Miyabe [32])**.** *An oracle x is low for uniform Schnorr tests if and only if for some (all) order function(s) h, every $f \leqslant_{\mathtt{tt}} x$ has a computable h-trace.*

Unfortunately, the usefulness of this approach is limited, because it is not the case that $\mathrm{NL}^{\mathtt{tt}}(A \star B) = \mathrm{NL}^{\mathtt{tt}}(A) \cup \mathrm{NL}^{\mathtt{tt}}(B)$. For suppose that $(a, F) \in (A \star B)_{\mathtt{inst}}$ is $\mathtt{tt}$-reducible to x, and has no computable $A \star B$-solution. Suppose further that $x \notin \mathrm{NL}^{\mathtt{tt}}(A)$; so a has a computable A-solution c. It is not necessarily the case that $F(c)$ is $\mathtt{tt}$-reducible to x. That is, we have a total computable function $I\colon 2^\omega \to \mathcal{F}$ such that $F = I(x)$. The function I gives us a *name* for F. But F itself is not continuous, only piecewise continuous, and the map taking $y \in 2^\omega$ to $I(y)(c)$ is not necessarily computable. When reducing to $A \star B$, we often describe non-continuous functions F, as they assume that the input is in fact a solution for the computed A-instance.

It would be interesting to find an extension of our methods that would allow us to characterise classes such as $\mathrm{NL}^{\mathtt{tt}}(\mathtt{Pass}(\mathscr{M}))$.

6. Addition, multiplication, and the Γ question

In this section we introduce another weakening of morphism reducibility (definition 6.9) that still implies cardinal inequality and containment of highness classes (lemma 6.10). This weakening is based on the dual operations of sum and product of Weihrauch problems, which have been used in both set theory and computable analysis.

We then use the new reducibility to present results from [34] in the language of morphisms (theorem 6.15). This result is closely related to Monin's resolution [33] of the Γ question which was stated in [1].

6.1. **Addition and multiplication.**

Definition 6.1. Let A and B be Weihrauch problems.

- The problem $A \times B$ is defined by letting $(A \times B)_{\mathtt{inst}} = A_{\mathtt{inst}} \times B_{\mathtt{inst}}$, $(A \times B)_{\mathtt{sol}} = A_{\mathtt{sol}} \times B_{\mathtt{sol}}$, and $(a, b)(A \times B)(\hat{a}, \hat{b})$ if and only if $aA\hat{a}$ *and* $bB\hat{b}$.

- The problem $A+B$ is defined by letting $(A+B)_{\mathtt{inst}} = A_{\mathtt{inst}} \times B_{\mathtt{inst}}$, $(A+B)_{\mathtt{sol}} = A_{\mathtt{sol}} \times B_{\mathtt{sol}}$, and $(a,b)(A+B)(\hat{a},\hat{b})$ if and only if $aA\hat{a}$ *or* $bB\hat{b}$.

Lemma 6.2. $A, B \to A \times B$.

Proof. To reduce A to $A \times B$, fix some computable $b^* \in B_{\mathtt{inst}}$. On the instance side map a to (a, b^*); on the solution side map $(\hat{a},\hat{b})$ to $\hat{a}$. Of course $B \to A \times B$ is identical. □

Remark 6.3. $A \times B \to A \star B$, but not always the other way round.

The following requires only running through the definitions:

Lemma 6.4.
(a) $(A \times B)^{\perp} = A^{\perp} + B^{\perp}$.
(b) $(A + B)^{\perp} = A^{\perp} \times B^{\perp}$.

As a result:

Lemma 6.5. $A + B \to A, B$.

It is also clear that the sum and product induce well-defined and nice operations on morphism classes, namely:

Lemma 6.6.
(a) If $A \to \hat{A}$ and $B \to \hat{B}$ then $A \times B \to \hat{A} \times \hat{B}$ and $A + B \to \hat{A} + \hat{B}$.
(b) $(A \times B) \times C \leftrightarrow A \times (B \times C)$, and $(A+B)+C \leftrightarrow A + (B+C)$.
(c) $A \times B \leftrightarrow B \times A$ and $A + B \leftrightarrow B + A$.

Lemma 6.7. Suppose that $\mathrm{Card}(A)$ and $\mathrm{Card}(B)$ are infinite. Then:
(a) $\mathrm{Card}(A \times B) = \max\{\mathrm{Card}(A), \mathrm{Card}(B)\}$; and
(b) $\mathrm{Card}(A + B) = \min\{\mathrm{Card}(A), \mathrm{Card}(B)\}$.

Proof. (a): By lemma 6.2, $\max\{\mathrm{Card}(A), \mathrm{Card}(B)\} \leqslant \mathrm{Card}(A \times B)$. To show equality, let Z be a complete solution set for A and let W be a complete solution set for B; then $Z \times W$ is a complete solution set for $A \times B$.

(b): By lemma 6.5, $\mathrm{Card}(A+B) \leqslant \min\{\mathrm{Card}(A), \mathrm{Card}(B)\}$. To show equality, let R be a complete solution set for $A+B$; then either the projection $\pi_0[R]$ is a complete solution set for A, or $\pi_1[R]$ is a complete solution set for B. □

Note how the proofs of lemma 6.7 are a simplification of the proofs of proposition 4.5(a) and proposition 4.6. Indeed we could deduce lemma 6.7 from these propositions, using the reduction $A \times B \to A \star B$.

Lemma 6.8.
(a) $\mathrm{H}(A \times B) = \mathrm{H}(A) \cap \mathrm{H}(B)$.
(b) $\mathrm{H}(A + B) = \mathrm{H}(A) \cup \mathrm{H}(B)$.

Proof. (a): By lemma 6.2, $\mathrm{H}(A \times B) \subseteq \mathrm{H}(A) \cap \mathrm{H}(B)$. For equality, let $x \in \mathrm{H}(A) \times \mathrm{H}(B)$; let $c \in A_{\mathtt{sol}}$ and $d \in B_{\mathtt{sol}}$ be both x-computable and solve every computable A-instance and every computable B-instance, respectively. Then (c,d) is x-computable and solves every computable $(A \times B)$-instance.

(b): By lemma 6.5, $\mathrm{H}(A) \cup \mathrm{H}(B) \subseteq \mathrm{H}(A+B)$. For equality, let $x \in \mathrm{H}(A+B)$; let (c,d) be x-computable and solve every computable $(A+B)$-instance. If c doesn't solve every computable A-instance, then d solves every computable B-instance. □

Using duality (lemma 6.4), we get the analogous results for the non-lowness classes: $\mathrm{NL}(A \times B) = \mathrm{NL}(A) \cup \mathrm{NL}(B)$ and $\mathrm{NL}(A + B) = \mathrm{NL}(A) \cap \mathrm{NL}(B)$.

These characterisations of cardinals and classes related to sums and products invite a weakening of morphism reduction. A *positive Boolean combination* of a problem A is a problem obtained from copies of A by repeatedly using addition and multiplication, for example $(A + A) \times (A + (A \times A))$. We can use these concepts to define yet another weakening of morphism implication by declaring all positive Boolean combinations of a problem to be equivalent:

Definition 6.9. Let R be the set of pairs (A, B) of Weihrauch problems such that $\hat{A} \to \hat{B}$ for some positive Boolean combination $\hat{A}$ of A and $\hat{B}$ of B. We write $A \rightsquigarrow B$ if (A, B) lies in the transitive closure of R.

The analysis so far yields the following:

Lemma 6.10. Suppose that $A \rightsquigarrow B$. Then:
- (a) $\mathrm{Card}(A) \leqslant \mathrm{Card}(B)$.
- (b) $\mathrm{H}(B) \subseteq \mathrm{H}(A)$.
- (c) $B^\perp \rightsquigarrow A^\perp$.

Part (c) uses the fact that the dual of a positive Boolean combination of A is a positive Boolean combination of $A^\perp$. It follows that if $A \rightsquigarrow B$ then $\mathrm{NL}(A) \subseteq \mathrm{NL}(B)$.

6.2. **Bounded IOE problems.** Bounded IOE problems were investigated in set theory by Kamo and Osuga [38] (this followed work on other cardinals indexed by growth rates of functions, for example [18, 23]). The associated highness classes in computability were introduced by Brendle and Nies [11].

For a function $h\colon \omega \to \omega$ we let $\mathrm{Bdd}(h) = h^\omega = \prod_n h(n)$ be the collection of functions $f\colon \omega \to \omega$ such that for all n, $f(n) < h(n)$. We let $\mathtt{IOE}(h)$ be the restriction of $\mathtt{IOE}$ to instances and solutions in $\mathrm{Bdd}(h)$. Throughout this section we assume that the bounding functions h are computable.

Lemma 6.11. If $\tilde{h} \leqslant h$ then $\mathtt{IOE}(\tilde{h}) \to \mathtt{IOE}(h)$.

Proof. Map an instance to itself; map a solution f to $\lambda n. \min\{f(n), \tilde{h}(n) - 1\}$. □

For functions h_0, h_1 we use the usual join operation: $(h_0 \oplus h_1)(2n + i) = h_i(n)$. For a function h, the splitting of h into the two standard columns is the pair (h_0, h_1) such that $h = h_0 \oplus h_1$.

Lemma 6.12. $\mathtt{IOE}(h_0) + \mathtt{IOE}(h_1) \to \mathtt{IOE}(h_0 \oplus h_1)$.

Proof. Map a problem (f_0, f_1) to the join $f_0 \oplus f_1$; map a solution $g = g_0 \oplus g_1$ to its splitting (g_0, g_1). If $(f_0 \oplus f_1)\mathtt{IOE}g$ then either $f_0\mathtt{IOE}g_0$ or $f_1\mathtt{IOE}g_1$. □

As a result, for every h, $\mathtt{IOE}(h) \rightsquigarrow \mathtt{IOE}(h \oplus h)$.

For a *computable real* number $a > 1$ we let

$$\ell_a(n) = \lfloor a^n \rfloor.$$

Lemma 6.13. Let $j(n)$ be a non-decreasing function. Then for all computable $a, b > 1$,

$$\mathtt{IOE}(j \circ \ell_b) \leftrightsquigarrow \mathtt{IOE}(j \circ \ell_a).$$

Proof. Let $b > 1$. Since j is non-decreasing, $(j \circ \ell_{b^2}) \oplus (j \circ \ell_{b^2}) \leqslant j \circ \ell_b$, and so by lemma 6.12 and lemma 6.11,

$$\mathtt{IOE}(j \circ \ell_{b^2}) \rightsquigarrow \mathtt{IOE}((j \circ \ell_{b^2}) \oplus (j \circ \ell_{b^2})) \to \mathtt{IOE}(j \circ \ell_b).$$

Iterating, for all $b > 1$ and $k \in \mathbb{N}$,

$$\mathtt{IOE}(j \circ \ell_{b^{2^k}}) \rightsquigarrow \mathtt{IOE}(j \circ \ell_b).$$

To prove the lemma, let $a, b > 1$. Without loss of generality $b < a$. For some k, $b^{2^k} > a$; by lemma 6.11,

$$\mathtt{IOE}(j \circ \ell_b) \to \mathtt{IOE}(j \circ \ell_a) \to \mathtt{IOE}(j \circ \ell_{b^{2^k}}) \rightsquigarrow \mathtt{IOE}(j \circ \ell_b). \qquad \square$$

Corollary 6.14. *For all computable $a > 1$, $\mathtt{IOE}(2^{2^n}) \leftrightsquigarrow \mathtt{IOE}(2^{\ell_a})$.*

6.3. **Besicovitch distance and the Γ question.** Recall that for finite binary strings σ, τ of the same length n, the normalised Hamming distance between σ and τ is

$$d(\sigma, \tau) = \frac{\#\{k < n \,:\, \sigma(k) \neq \tau(k)\}}{n}.$$

For $x, y \in 2^\omega$, the Besicovitch distance between x and y is

$$d(x, y) = \limsup_n d(x \upharpoonright n, y \upharpoonright n).$$

Brendle and Nies [11], motivated by [22], introduced the following Weihrauch problems, for each computable $p \in [0, 1]$:

- $\mathtt{Close}(p)$: an instance is $x \in 2^\omega$; a solution is $y \in 2^\omega$ such that $d(x, y) < p$.
- The dual of $\mathtt{Close}(p)$ is $\mathtt{Far}(p)$: an instance is $x \in 2^\omega$; a solution is $y \in 2^\omega$ such that $d(x, y) \geqslant p$.

Thus, $\mathrm{H}(\mathtt{Close}(p))$ is the set of oracles that compute a point which has distance $< p$ to any computable point; this is clearly nonempty only if $p > 1/2$. On the other hand, $\mathrm{NL}(\mathtt{Close}(p)) = \mathrm{H}(\mathtt{Far}(p))$ is the set of oracles that compute a point which has distance at least p from any computable point. See [37] for a formalisation using Hausdorff distance.

Hirschfeldt et al. [22] showed that for positive $p \leqslant 1/2$, $\mathrm{NL}(\mathtt{Close}(p))$ consists of all of the nonzero degrees, so again we are interested in the case $p > 1/2$. Brendle and Nies [11] examined relationships between the associated cardinals and highness classes, and their relationship with the Cichoń diagram. Monin and Nies [35, 34] then showed that the highness classes and cardinals associated with the problems $\mathtt{Close}(p)$ and $\mathtt{IOE}(2^{2^n})$ are related. In morphism form, their results give:

Theorem 6.15. *For all computable $p \in (1/2, 1)$,*

$$\mathtt{Far}(p) \leftrightsquigarrow \mathtt{IOE}(2^{2^n}).$$

It follows, of course, that for all computable $p, q \in (1/2, 1)$, $\mathtt{Far}(p) \leftrightsquigarrow \mathtt{Far}(q)$ (and so also that $\mathtt{Close}(p) \leftrightsquigarrow \mathtt{Close}(q)$). As a result $\mathrm{H}(\mathtt{Close}(p)) = \mathrm{H}(\mathtt{Close}(q))$ and $\mathrm{H}(\mathtt{Far}(p)) = \mathrm{H}(\mathtt{Far}(q))$, and $\mathrm{Card}(\mathtt{Close}(p)) = \mathrm{Card}(\mathtt{Close}(q))$, which is how the result is stated in [34] (which also investigated the amount of uniformity of the reductions). Monin and Nies's work was partly motiveated by the so-called Γ *question.*

For $x \in 2^\omega$, consider

$$d(x, \mathcal{R}) = \inf \{d(x, y) \,:\, y \text{ is computable}\}.$$

Andrews et al. [1] define

$$\Gamma(\mathbf{d}) = 1 - \sup_{x\in\mathbf{d}} d(x,\mathcal{R}).$$

So $1-\Gamma(\mathbf{d})$ says how far we can get from computable points among the **d**-computable points. The values $0, 1$, and $1/2$ are possible; their observation above shows that values strictly between $1/2$ and 1 are not. Andrews et al. [1] asked whether values between 0 and $1/2$ are possible. Monin showed:

Theorem 6.16 (Monin)**.** *If* $\Gamma(\mathbf{d}) < 1/2$ *then* $\Gamma(\mathbf{d}) = 0$*.*

Theorem 6.15 gives a quick proof of Monin's result:

Proof. Let $y \in \mathbf{d}$ and suppose that $d(y,\mathcal{R}) = p > 1/2$. Let $q \in (1/2, 1)$ be computable; we need to show that there is some $x \leqslant_{\mathrm{T}} y$ such that $d(x,\mathcal{R}) \geqslant q$; this follows from $\mathrm{H}(\mathtt{Far}(\tilde{p})) = \mathrm{H}(\mathtt{Far}(q))$, where $\tilde{p} \in (1/2, p)$ is computable. □

This is a-historical: In [35], Monin and Nies first showed that $\mathtt{Close}(p)$ and $\mathtt{IOE}(2^{2^n})$ are related; they essentially proved one direction of theorem 6.15 (which was later stated in morphism form by Kihara [24]). Monin [33] then used the list decoding theorem, discussed below, to prove theorem 6.16; this technique was then used by Monin and Nies in [34] to prove the other direction of theorem 6.15. We now present the proof of theorem 6.15 in morphism form.

The main map. For a function $\ell\colon \omega \to \omega$ we let $\mathrm{StrLth}(\ell) = \prod_n \{0,1\}^{\ell(n)}$ be the collection of functions $f\colon \omega \to 2^{<\omega}$ such that for all n, $|f(n)| = \ell(n)$. This collection is naturally identified with $\mathrm{Bdd}(2^\ell)$.

For a function $\ell\colon \omega \to \omega$ we define a bijection $\Phi_\ell\colon 2^\omega \to \mathrm{StrLth}(\ell)$: Let $\langle J^\ell_n \rangle$ be the partition of ω into an increasing sequence of intervals, with $|J_n| = \ell(n)$. Then $\Phi_\ell(x)(n) = x\restriction J_n$. The inverse of Φ_ℓ is obtained by concatenating the values of a function in $\mathrm{StrLth}(\ell)$. In this section, all the functions ℓ are computable.

For the following lemma, and below, for $x, y \in 2^\omega$, we let

$$\underline{d}(x,y) = \liminf_n d(x\restriction n, y\restriction n).$$

Lemma 6.17. Let $a > 1$, and let $x, y \in 2^\omega$; let $r \in [0,1]$.

(a) Suppose that $d(\Phi_{\ell_a}(x)(n), \Phi_{\ell_a}(y)(n)) \leqslant r$ for infinitely many n. Then $\underline{d}(x,y) \leqslant 1/a + r(1-1/a)$.

(b) Suppose that $d(\Phi_{\ell_a}(x)(n), \Phi_{\ell_a}(y)(n)) \leqslant r$ for all but finitely many n. Then $d(x,y) \leqslant r + (a-1)$.

Proof. Let $\langle J_n \rangle = \langle J^{\ell_a}_n \rangle$ be the increasing sequence of intervals with $|J_n| = \ell_a(n)$, and let $b_n = \max J_n = \sum_{m\leqslant n} \ell_a(m)$. For brevity, let $x_n = x\restriction J_n = \Phi_{\ell_a}(x)(n)$ and similarly let $y_n = y\restriction J_n$.

Suppose that $d(x_n, y_n) \leqslant r$. Then

$$d(x\restriction b_n, y\restriction b_n) \leqslant \frac{b_{n-1}}{b_n} + r\frac{\ell_a(n)}{b_n}.$$

For (a), we then use the fact that $\lim_n b_{n-1}/b_n = 1/a$ and $\lim_n \ell_a(n)/b_n = 1 - 1/a$.

For (b), as distance is invariant under finite changes, we may assume that for all n, $d(x_n, y_n) \leqslant r$. Let $n < \omega$ and let $m \in J_n$. Then

$$d(x\restriction m, y\restriction m) \leqslant \frac{rb_{n-1} + (m - b_{n-1})}{m} \leqslant \frac{rb_{n-1} + \ell_a(n)}{b_{n-1}} = r + \frac{\ell_a(n)}{b_{n-1}},$$

and note that $\lim_n \ell_a(n)/b_{n-1} = a - 1$. □

Proof of one direction of theorem 6.15: $\texttt{Far}(p) \rightsquigarrow \texttt{IOE}(2^{2^n})$. This of course will work for all $p \in (0,1)$. Even though $\underline{d}$ defined above is not a pseudo-metric, we define, for $q \in [0,1]$, the analogous Weihrauch problem, $\underline{\texttt{Close}}(q)$: an instance is $x \in 2^\omega$; a solution is $y \in 2^\omega$ such that $\underline{d}(x,y) \leqslant q$. Note that we use a non-strict inequality. This is so that

$$\texttt{Far}(p) \leftrightarrow \underline{\texttt{Close}}(1-p).$$

To see this, we observe that for strings σ and τ of the same length, $d(\sigma^\complement, \tau) = 1 - d(\sigma,\tau)$ and so for $x, y \in 2^\omega$, $d(x^\complement, y) = 1 - \underline{d}(x,y)$. The morphisms take a complement on the instance side and use the identity for solutions.

It remains, therefore, to show that $\underline{\texttt{Close}}(q) \rightsquigarrow \texttt{IOE}(2^{2^n})$ for all computable $q \in (0,1)$. Given such q, let $a = 1/q$, so $a > 1$. Using Φ_{ℓ_a} on the instance side, and $\Phi_{\ell_a}^{-1}$ on the solution side, we get

$$\underline{\texttt{Close}}(q) \to \texttt{IOE}(2^{\ell_a}).$$

This follows from lemma 6.17(a), with $r = 0$. The proof then ends by quoting corollary 6.14. □

As discussed above, this direction was proved in [35]; the morphism $\underline{\texttt{Close}}(q) \to \texttt{IOE}(2^{\ell_{1/q}})$ was explicitly stated by Kihara [24, Prop.3.8(1)].

Infinitely often closeness. Toward the other direction, we introduce along the way a few intermediate problems. Again $\ell\colon \omega \to \omega$ is a function and $p \in [0,1]$ is a real number. We consider the following Weihrauch problem:

- $\texttt{IOClose}(\ell, p)$: an instance is $f \in \mathrm{StrLth}(\ell)$; a solution is $g \in \mathrm{StrLth}(\ell)$ such that for infinitely many n, $d(f(n), g(n)) \leqslant p$.

Lemma 6.18. Let $r \in [0,1]$, let $a > 1$, and let $p \in [0,1]$ such that $p > r + (a-1)$. Then

$$\texttt{IOClose}(\ell_a, 1-r) \to \texttt{Far}(p).$$

Proof. We verify the dual: $\texttt{Close}(p) \to \texttt{AEFar}(\ell_a, 1-r)$, where $g \in \mathrm{Bdd}(\ell)$ is a $\texttt{AEFar}(\ell, q)$-solution for f if for all but finitely many n, $d(f(n), g(n)) \geqslant q$. By mapping, on the instance side, a function $f \in \mathrm{StrLth}(\ell)$ to $\lambda n.f(n)^\complement$, we see that $\texttt{AEFar}(\ell, 1-r)$ is equivalent to $\texttt{AEClose}(\ell, r)$, where as expected a solution g now staisfies, for all but finitely many n, $d(f(n), g(n)) < r$.

To reduce $\texttt{Close}(p) \to \texttt{AEFar}(\ell_a, r)$ we use Φ_{ℓ_a} on the instance side and $\Phi_{\ell_a}^{-1}$ on the solution side, appealing to lemma 6.17(b). □

Bounded traces. Onwards to tracing. We look at bounded traces. Again fix ℓ; the functions to be traced are still elements of $\mathrm{StrLth}(\ell)$. As in section 2, we name traces by effectively coding finite sets by natural numbers. We now deal with traces that are bounded by a constant function. The Weihrauch problem is:

- $\texttt{IOTrace}(\ell, L)$: an instance is a function $f \in \mathrm{StrLth}(\ell)$. A solution is an L-trace T which infinitely often traces f.

The list decoding theorem says: for all $q \in (0, 1/2)$ there are $L = L(q)$ and $\varepsilon = \varepsilon(q)$ such that for all m, there is a set $C_m \subseteq \{0,1\}^m$ of size $2^{\lfloor \varepsilon m \rfloor}$ which is sparse in the sense that every string $\sigma \in \{0,1\}^m$ is within distance $< q$ to at most L elements of C_m.

We now fix $q \in (0, 1/2)$ and obtain the resulting ε and L.

Lemma 6.19. For all $\ell\colon \omega \to \omega$,

$$\texttt{IOTrace}(\lfloor \varepsilon\ell \rfloor, L) \to \texttt{IOClose}(\ell, q).$$

Proof. Fix $\langle C_m \rangle$ as promised. Since $|C_m| = 2^{\lfloor \varepsilon m \rfloor}$, we identify $\mathrm{StrLth}(\lfloor \varepsilon\ell \rfloor)$ with $\prod_n C_{\ell(n)}$, which is a subset of $\mathrm{StrLth}(\ell)$.

So now on the instance side we can take the identity function. Given a solution $g \in \mathrm{StrLth}(\ell)$, let $T(n) = \{\sigma \in C_{\ell(n)} : d(\sigma, g(n)) < q\}$. The properties of C_m mean that $|T(n)| \leqslant L$. If g is an $\texttt{IOClose}(\ell, q)$-solution for f then T is an $\texttt{IOTrace}(\lfloor \varepsilon\ell \rfloor, L)$-solution for f. □

Lemma 6.20. For all non-decreasing ℓ and L,

$$\texttt{IOTrace}(\ell, L) \leftrightsquigarrow \texttt{IOTrace}(\ell \oplus \ell, L).$$

Proof. Similar to the proof of lemma 6.12. If $\tilde{\ell} \leqslant \ell$ then as in lemma 6.11, $\texttt{IOTrace}(\tilde{\ell}, L) \to \texttt{IOTrace}(\ell, L)$. On the instance side, map f to the function g obtained by extending each string $f(n)$ (of length $\tilde{\ell}(n)$) to a string of length $\ell(n)$, say by adding zeros; map a solution T to $\lambda n.\{\sigma\restriction\tilde{\ell}(n) : \sigma \in T(n)\}$.

Next we see that $\texttt{IOTrace}(\ell_0, L) + \texttt{IOTrace}(\ell_1, L) \to \texttt{IOTrace}(\ell_0 \oplus \ell_1, L)$. On the instance side, we map (f, g) to $f \oplus g$; on the solution side, we map T to the pair (T_0, T_1), where $T_i(n) = T(2n + i)$. □

Corollary 6.21. *For all $a > 1$, $\varepsilon > 0$ and L,*

$$\texttt{IOTrace}(L \cdot 2^n, L) \leftrightsquigarrow \texttt{IOTrace}(\lfloor \varepsilon\ell_a \rfloor, L).$$

Proof. This is an elaboration on the proof of lemma 6.13, this time adding multiplicative constants. The proof of lemma 6.13, this time using lemma 6.20, shows that for all computable $a, b > 1$, $\texttt{IOTrace}(\ell_a, L) \leftrightsquigarrow \texttt{IOTrace}(\ell_b, L)$. For any computable $\alpha > 0$ and $b > 1$, $\alpha\ell_b$ is dominated by ℓ_a for all $a > b$ and dominates ℓ_a for all $a < b$. Since we only need to correctly trace infinitely often, the morphisms can ignore finitely many "wrong" n's. □

The last step is:

Lemma 6.22. For all ℓ and L, $\texttt{IOE}(2^\ell) \rightsquigarrow \texttt{IOTrace}(L \cdot \ell, L)$.

Proof. We prove that $\sum_L \texttt{IOE}(2^\ell) \to \texttt{IOTrace}(L \cdot \ell, L)$. We think of $\mathrm{Bdd}(2^\ell)$ as $\mathrm{StrLth}(\ell)$. Map $f \in \mathrm{StrLth}(\ell)$ to the function mapping n to the conctenation of $f(n)$ with itself L times. Now let T be a trace; we may assume each $T(n)$ contains exactly L elements. for $i < L$ let $g_i(n)$ be the i^{th} block of the i^{th} element of $T(n)$. For some i, for infinitely many n, $f(\sigma)\hat{}\cdots\hat{}f(\sigma)$ equals the i^{th} element of $T(n)$; for that i, $f\,\texttt{IOE}\,g_i$. □

Proof of theorem 6.15. Let $p \in (1/2, 1)$; it remains to show that $\texttt{IOE}(2^{2^n}) \rightsquigarrow \texttt{Far}(p)$. Pick some $r \in (1/2, p)$; pick some $a > 1$ such that $a - 1 < p - r$. So $p > r + (a - 1)$. Let $q = 1 - r$ and let $\varepsilon = \varepsilon(q)$ and $L = L(q)$ be obtained from the list decoding theorem.

By lemma 6.22, $\texttt{IOE}(2^{2^n}) \rightsquigarrow \texttt{IOTrace}(L \cdot 2^n, L)$. By corollary 6.21, $\texttt{IOTrace}(L \cdot 2^n, L) \rightsquigarrow \texttt{IOTrace}(\lfloor \varepsilon\ell_a \rfloor, L)$. By lemma 6.19, $\texttt{IOTrace}(\lfloor \varepsilon\ell_a \rfloor, L) \to \texttt{IOClose}(\ell_a, q)$. By lemma 6.18, $\texttt{IOClose}(\ell_a, q) \to \texttt{Far}(p)$. □

7. On Kurtz

The σ-ideal $\mathscr{E}$ generated by compact null sets was studied by Bartozysnki and Shelah, who characterised, among others, the cardinals $\mathbf{add}(\mathscr{E})$ and $\mathbf{cof}(\mathscr{E})$. In computability, the associated notion of randomness / genericity is that of *Kurtz randomness*: avoiding all null, effectively closed subsets of Cantor space. The relevant Weihrauch problem is $\texttt{Supset}(\mathscr{E})$: an instance is a null $\mathbf{\Sigma}^0_2$ set P; a solution is a null $\mathbf{\Sigma}^0_2$ set Q such that $P \subseteq Q$.

Our goal in this section is:

Theorem 7.1. $\texttt{Supset}(\mathscr{E}) \leftrightarrow \texttt{Supset}(\mathscr{M})$.

As a result we get:

Theorem 7.2 (Bartoszynski, Shelah [5])**.** $\mathbf{add}(\mathscr{E}) = \mathbf{add}(\mathscr{M})$ *and* $\mathbf{cof}(\mathscr{E}) = \mathbf{cof}(\mathscr{M})$.

Together with Corollary 4.15, we obtain:

Theorem 7.3. *A degree is low for $\mathbf{\Sigma}^0_2$ null sets if and only if it is hyperimmune-free and not* DNR.

7.1. **Lowness for Kurtz tests and randomness.** As with the analogus problem $\texttt{Supset}(\mathscr{M})$, in computability, the notions which have been studied are not lowness for $\mathbf{\Sigma}^0_2$ null sets, but rather, lowness for closed null sets (lowness for Kurtz test), and lowness for Kurtz randomness: computing a closed, null set Q which contains a Kurtz random point. The method we used above of proving equivalence of all three notions can be modified to the Kurtz context as well.

Let $\mathscr{KN}$ be the collection of closed, null sets. First, we obtain the analogue of proposition 4.16:

Proposition 7.4. $\texttt{Supset}(\mathscr{E}) \to \texttt{Supset}(\mathscr{KN})$.

The proof is identical. We then define the problem $\texttt{FractalSpill}(\mathscr{KN})$ as above.

Proposition 7.5. $\texttt{Spill}(\mathscr{KN}) \leftrightarrow \texttt{FractalSpill}(\mathscr{KN})$.

Proof. The construction is identical to that proving lemma 4.18, except that having determined that $n \in I$, for $\sigma \in S$ of length n, we observe that under the assumption that $[T]$ is null, the tree $\bigcup_{k \leqslant |\sigma|} (\sigma \upharpoonright k)\hat{\ } T$ is null as well, and so we can find a clopen set $D \subseteq [\sigma]$ disjoint from that tree such that $\lambda(D|\sigma) \geqslant 1 - 2^{-n}$; we remove D from S. The next element of I is bigger than the granularity of all such D's. $\square$

The rest is identical: if Γ is a countable collection of closed, null sets, closed under the shift operator, and (Q, I) is a $\texttt{FractalSpill}(\mathscr{KN})$-solution for all $P \in \Gamma$, then we build a point in Q avoiding all $P \in \Gamma$. We thus obtain:

Theorem 7.6 (Greenberg, J. Miller [19])**.** *The following are equivalent for $x \in 2^\omega$:*

1. *x is low for Kurtz tests.*
2. *x is low for Kurtz randomness.*
3. *x is hyperimmune-free and not* DNR.

We also obtain the analogous result in the Δ^1_1 context:

Theorem 7.7 (Kjos-Hanssen, Nies, Stephan, Yu [25])**.** *An oracle is low for Δ^1_1 Kurtz randomness if and only if it is Δ^1_1-dominated and every $f \in \Delta^1_1(x)$ is infinitely often equal to some Δ^1_1 function.*

7.2. **Revising the problem.** Our goal, as stated, is theorem 7.1. It turns out, though, that the bulk of our analysis concerns the following Weihrauch problem:

- $\mathtt{Supset}(\mathscr{E},\mathscr{N})$: an instance is a null $\mathbf{\Sigma}^0_2$ set P; a solution is a null set R such that $P \subseteq R$.

The identity map in both directions shows that $\mathtt{Supset}(\mathscr{E},\mathscr{N}) \to \mathtt{Supset}(\mathscr{E})$. We will show:

Proposition 7.8. $\mathtt{Supset}(\mathscr{E},\mathscr{N}) \longleftrightarrow \mathtt{IOE}$*.*

As a result, we obtain:

Theorem 7.9 (Greenberg, J. Miller [19])**.** *Let $x \in 2^\omega$. There is an x-computable $\mathbf{\Sigma}^0_2$ null set not contained in any Schnorr null set if and only if x is high or* DNR*.*

Before we prove proposition 7.8, we show how it implies the main theorem. We use the following two reductions.

Proposition 7.10. $\mathtt{Dom} \to \mathtt{Supset}(\mathscr{E})$*.*

Proof. On the instance side, given $f \in \omega^\omega$, we assume that f is strictly increasing, and map it to

$$Q^f = \{x \in 2^\omega \,:\, (\forall n)\, x(f(n)) = 0\}\,.$$

On the solution side, we are given a sequence $\langle T_n\rangle$ of closed null sets. Let $h(n,k)$ be the least s such that every string σ of length k has an extension of length s which is off T_n. Let $g(0) = 0$ and $g(n+1) = h(g(n), g(n))$; we map $\langle T_n\rangle$ to $n \mapsto g(2n)$.

To show that this works, suppose that $f \in \omega^\omega$, and that for some $\langle T_n\rangle$, the function $n \mapsto g(2n)$ does not dominate f. We claim that there are infinitely many n for which for some m, $f(n) \leqslant g(m) \leqslant g(m+1) < f(n+1)$. Otherwise, for some k, for large enough n, $f(n) \leqslant g(n+k) \leqslant g(2n)$. We use this to construct a point $x \in Q^f \backslash \bigcup_n [T_n]$. As usual, we assume that $\langle T_n\rangle$ is nested, so it suffices to avoid infinitely many trees T_n. To do this, suppose that we have already determined some finite $\sigma \prec x$ such that $[\sigma] \cap Q^f \neq \varnothing$ of length some $f(n)$ such that $f(n) \leqslant g(m) \leqslant g(m+1) < f(n+1)$. Then for every extension τ of σ of length $g(m+1)$, $[\tau] \cap Q^f \neq \varnothing$. On the other hand, we can find such an extension of σ which is off $T_{g(m)}$. □

Remark 7.11*.* The proof actually shows that $\mathtt{Dom} \to \mathtt{Supset}(\mathscr{E},\mathscr{M})$.

Proposition 7.12. $\mathtt{Supset}(\mathscr{E}) \to \mathtt{Supset}(\mathscr{E},\mathscr{N}) \star \mathtt{Dom}$*.*

Proof. This resembles the proof of proposition 4.9, with a dose of compactness.

On the instance side, map a $\mathbf{\Sigma}^0_2$ null set $P = \bigcup_n P_n$ to the pair (P,F) where for a null set V (given as $\bigcap_m V_m$) such that $P \subseteq V$, we let $F(V)$ be the function $h \in \omega^\omega$ defined as follows: $h(n,m)$ is the least s such that $P_{n,s} \subseteq V_{m,s}$. (The closed sets P_n are given as trees T_n, and we let $P_{n,s}$ be the clopen set determined by the s^{th} level of T; we similarly treat each V_m.) By compactness, h is well-defined, and in fact is uniformly computable given P and V.

On the solution side, we map a pair (V,g) to the $\mathbf{\Sigma}^0_2$ null set $\bigcup_n \bigcap_m V_{m,g(n,m)}$. To show that this works: if $g(n,-)$ majorises $h(n,-)$ (where $h = F(V)$) then $P_n \subseteq \bigcap_m V_{m,g(n,m)}$; as usual we may assume that $P_n \subseteq P_{n+1}$. □

Proof of theorem 7.1, assuming proposition 7.8. In one direction, we first observe that $\mathtt{Pass}(\mathscr{M}) \dashrightarrow \mathtt{Supset}(\mathscr{E})$; this is because

$$\mathtt{Pass}(\mathscr{M}) \dashrightarrow \mathtt{IOE} \dashrightarrow \mathtt{Supset}(\mathscr{E},\mathscr{M}) \to \mathtt{Supset}(\mathscr{E}).$$

Together with proposition 7.10 (and using the fact that $A \dashrightarrow C, B \dashrightarrow C$ implies $A \star B \dashrightarrow C$) we get

$$\mathtt{Supset}(\mathscr{M}) \to \mathtt{Pass}(\mathscr{M}) \star \mathtt{Dom} \dashrightarrow \mathtt{Supset}(\mathscr{E}).$$

In the other direction, if $\mathtt{Supset}(\mathscr{E},\mathscr{N}) \to \mathtt{Pass}(\mathscr{M})^{\star k}$, then using proposition 7.12,

$$\mathtt{Supset}(\mathscr{E}) \to \mathtt{Pass}(\mathscr{M})^{\star k} \star \mathtt{Dom} \to \mathtt{Supset}(\mathscr{M})^{\star(k+1)}.$$ □

7.3. **Analysis of $\mathtt{Supset}(\mathscr{E},\mathscr{N})$.** We work toward a proof of proposition 7.8.

Proposition 7.13. $\mathtt{Supset}(\mathscr{E},\mathscr{N}) \to \mathtt{IOE}$.

Proof. On the instance side, we are given a $\mathbf{\Sigma}^0_2$ null set $P = \bigcup P_n$. We map it to a sequence $\langle C_n \rangle$ of (codes of) clopen sets such that for all n, $P_n \subseteq C_n$ and $\lambda(C_n) = 2^{-n}$. (For each n, find a late enough $s(n)$ and let $C_n = T_{n,s(n)}$.)

On the solution side, we are given a sequence $\langle D_n \rangle$ of clopen sets, and we may assume that for all n, $\lambda(D_n) = 2^{-n}$. We map this sequence to the null set $\bigcap_k U_k$ given by $U_k = \bigcup_{n>k} D_n$. The measure of each U_k is computable from $\langle D_n \rangle$, uniformly. If $\langle P_n \rangle$ maps to $\langle C_n \rangle$ and for infinitely many n, $D_n = C_n$ then for each n and k, $P_n \subseteq U_k$. □

Proposition 7.14. $\mathtt{Esc} \to \mathtt{Supset}(\mathscr{E},\mathscr{N})$.

Proof. The following argument resembles the proof of [31, Thm.2.2]; the argument of the corresponding cardinal inequality in [4] (Lemma 2.6.13) is non-constructive.

As above, given $f \in \omega^\omega$, assuming it is increasing, define

$$Q^f = \{x \in 2^\omega : (\forall n)\, x(f(n)) = 0\}.$$

The map $f \mapsto Q^f$ is our map on instances.

Toward defining our map on solutions, let, for $k < \omega$, $q_k = (3/4)2^{-2k}$. What we use[11] is:

$$\sum_{k=0}^{\infty} q_k = 1,$$

and

$$\sum_{k=0}^{\infty} q_k 2^k < 2.$$

We are given a null set $V = \bigcap_m V_m$. As mentioned above, by [15, Prop.7.1.6], we assume that for each m, $\lambda(V_m) = 2^{-m}$. As above we use the clopen approximations Q^f_s and $V_{m,s}$, determined by sets of strings of length s.

For each $m \geqslant 1$, define a function $h_m \colon [m-1,\omega) \to \omega$ by letting, for $n \geqslant m-1$,

$$h_m(n) = (\mu s)\, \lambda(V_{m,s} | V_m) \geqslant \sum_{k \leqslant n} q_{k-(m-1)}.$$

[11] Also of course we use the fact that $\langle q_k \rangle$ is computable and consists of binary rationals. Below we will also assume that $f(k)$ is at least the granularity of q_k, i.e. $f(k) \geqslant 2k+2$.

On the solution side, we map V to the function

$$h(n) = \max_{m \leqslant n+1} h_m(n).$$

Thus for all $m \geqslant 1$ and $n \geqslant m-1$, $h_m(n) \leqslant h(n)$.

To show that this works, we need to show that if $Q^f \subseteq V$ then h escapes f.

The rough idea is that if h is majorised by f beyond $m-1$, then V_m is spending its measure before Q^f has thinned itself (by committing to the bit at position $f(n)$ being 0). Since the later thinnings will happen independently of this preemptively spent measure, most of this measure is wasted, and so V_m will not have enough measure to cover Q^f.

The following claim is the combinatorial heart of this proof.

Claim 7.14.1. Fix $m \geqslant 1$ and $d \geqslant m-1$. Suppose that for all $n \in [m-1, d]$, $h_m(n) \leqslant f(n)$. Then $Q^f_{f(d)} \nsubseteq V_{m,f(d)}$.

The claim gives the proposition. To see this, suppose that $Q^f \subseteq V$. Then $Q^f \subset V_m$ for every m. By compactness, for every m there is some d such that $Q^f \subset V_{m,f(d)}$. Because every string of length $f(d)$ on (the tree defining) Q^f is extendible, we in fact have $Q^f_{f(d)} \subseteq V_{m,f(d)}$. The claim then ensures that for every m there is some $n \geqslant m-1$ such that $h_m(n) > f(n)$. So $h(n) \geqslant h_m(n) > f(n)$.

It thus remains to prove the claim.

By definition of h_m, and under the assumption of the claim, for all $n \in [m-1, d]$ we have

$$\lambda(V_{m,f(n)} | V_m) \geqslant \sum_{k=0}^{n-(m-1)} q_k.$$

By removing short strings from the set of strings defining V_m, but leaving their extensions, we may, without decreasing $V_{m,f(d)}$, assume that we have equality:

$$\lambda(V_{m,f(n)} | V_m) = \sum_{k=0}^{n-(m-1)} q_k.$$

Define $U_{m-1} = V_{m,f(m-1)}$ and for $n \in [m, d]$ let

$$U_n = V_{m,f(n)} - V_{m,f(n-1)}.$$

So for all $n \leqslant d$, $\lambda(U_n) = q_{n-(m-1)} \lambda(V_m) = q_{n-(m-1)} 2^{-m}$.

Fix $n \in [m-1, d]$. The set U_n can only have "opinions" about the bits $f(r)$ for $r < n$, while those bits $f(r)$ with $r \in [n, d)$ are independent of U_n. So

$$\lambda\big(U_n \cap Q^f_{f(d)}\big) = \lambda\big(U_n \cap Q^f_{f(n)}\big) \cdot 2^{-(d-n)} \leqslant \lambda(U_n) \cdot 2^{-(d-n)} = q_{n-(m-1)} 2^{-m} 2^{-(d-n)}.$$

It follows that

$$\lambda\big(V_{m,f(d)} \cap Q^f_{f(d)}\big) = \sum_{n=m-1}^{d} \lambda\big(U_n \cap Q^f_{f(d)}\big) \leqslant$$
$$2^{-d} \sum_{n=m-1}^{d} q_{n-(m-1)} 2^{n-m} < 2^{-d} \cdot \frac{1}{2} \sum_{k=0}^{\infty} q_k 2^k < 2^{-d},$$

whereas $\lambda(Q^f_{f(d)}) = 2^{-d}$. Hence it is impossible that $Q^f_{f(d)} \subseteq V_{m,f(d)}$. □

Proposition 7.15. $\mathtt{IOE} \to \mathtt{Supset}(\mathscr{E}, \mathscr{N}) \star \mathtt{Esc}$.

Proof. We use the "Svelte tree" machinery from [19]. Let $0 = n_0 < n_1 < n_2 \dots$ be a computable increasing sequence such that $n_k - n_{k-1} = 2^{2k}$. Fix a computable bijection $I\colon \omega^{<\omega} \to \omega$. For $f \in \omega^\omega$ we define

$$P^f = \{x \in 2^\omega \ : \ (\forall k)\, x(I(f \upharpoonright n_k)) = 0\}\,.$$

Let $k < \omega$. A finite tree $T \subset \omega^{<\omega}$ is *k-svelte* if there are finite sets of sequences $\langle S_{k+1}, S_{k+2}, S_{k+3}, \dots\rangle$ such that:

- for all $m \geqslant k+1$, $S_m \subseteq T$ and every $\sigma \in S_m$ has length n_m;
- $|S_m| \leqslant 2^m$;
- Every leaf of T extends a sequence in $\bigcup_{m \geqslant k+1} S_m$.

In fact, we can get $|S_m| \leqslant 2^{m-(k+1)}$, but we don't need this. The main combinatorial result of [19] is:

Theorem 7.16 (Thm. 3.3 of [19])**.** *Let $C \subset 2^\omega$ be clopen, and suppose that $\lambda(C) \leqslant 2^{-(k+1)}$. Then there is a k-Svelte tree T such that for all $f \in \omega^\omega$, if $P^f \subset C$ then f extends some leaf of T. The tree T can be obtained computably from C.*

The map on instances is as follows. We map $f \in \omega^\omega$ to the pair (P^f, F), where for a null set $V = \bigcap_k V_k$ such that $P^f \subseteq V$, $F(V)$ is the function h mapping $k < \omega$ to a stage s such that $P^f_s \subseteq V_{k,s}$.

On the solution side, we map a pair $(V, \hat{h}) \in \mathscr{N} \times \omega^\omega$ to a point $g \in \omega^\omega$ defined as follows. For each $k \geqslant 1$ we compute a $(k-1)$-Svelte tree T_k such that for every $\tilde{f} \in \omega^\omega$, if $Q^{\tilde{f}} \subset V_{k,\hat{h}(k)}$ then $\tilde{f}$ extends a leaf of T_k. Let $S^k_k, S^k_{k+1}, S^k_{k+2}, \dots$ witness that T_k is $(k-1)$-svelte. For each $m \geqslant 1$ let $S_m = \bigcup_{k \leqslant m} S^k_m$. So $|S_m| \leqslant 2^{2m}$. Since $I_m = [n_{m-1}, n_m)$ has size 2^{2m}, we can pick injectively, for each $\sigma \in S_m$, a value $l = l(\sigma) \in I_m$. We define g so that $g(l) = \sigma(l)$. So we have arranged that for every k, for every leaf τ of T_k, τ agrees with g on some value $l \geqslant n_{k-1}$.

Suppose that $P^f \subseteq V$ and that $\hat{h}(k) \geqslant h(k)$. Then $P^f \subset V_{k,\hat{h}(k)}$. It follows that f extends some leaf of T_k, and therefore that g agrees with f at some value $l \geqslant n_{k-1}$. Therefore, if $\hat{h}$ escapes h, then $f\,\mathtt{IOE}\,g$. ☐

Proof of proposition 7.8. By proposition 7.13, $\mathtt{Supset}(\mathscr{E}, \mathscr{N}) \to \mathtt{IOE}$. In the other direction, we have $\mathtt{IOE} \to \mathtt{Supset}(\mathscr{E}, \mathscr{N}) \star \mathtt{Esc}$ and $\mathtt{Esc} \to \mathtt{Supset}(\mathscr{E}, \mathscr{N})$, so $\mathtt{IOE} \to \mathtt{Supset}(\mathscr{E}, \mathscr{N})^{\star 2}$. ☐

7.4. **A self-contained argument.** The drawback of the proof given above of $\mathtt{IOE} \dashrightarrow \mathtt{Supset}(\mathscr{E}, \mathscr{N})$ is its reliance on the Svelte tree technology of [19], which takes some work. The original argument is essentially due to Baroszynski and Shelah. We use their argument (simplified at a few places) to give a self-contained proof of this implication. This proof uses techniques which may be of independent interest. We start with a fact which we will use twice. We define the Weihrauch problem:

- $\mathtt{Esc}\neg\mathtt{Dom}$: an instance is a function $f \in \omega^\omega$ such that $f(n) > n$; a solution is a non-decreasing function $g \in \omega^\omega$, which escapes f but does not dominate it (and $g(n) \geqslant n$).

Lemma 7.17 (With Harrison-Trainor). $\mathtt{Esc} \leftrightarrow \mathtt{Esc}\neg\mathtt{Dom}$.

Proof. To reduce $\texttt{Esc} \to \texttt{Esc}\neg\texttt{Dom}$, we map an instance f to $n \mapsto \max\{f(n), n+1\}$, and use the identity on the solutions.

We reduce $\texttt{Esc}\neg\texttt{Dom}$ to $\texttt{Esc}$. We first define h by $h(0) = f(0)$ and $h(n+1) = \max\{f(n+1), h(n)+1\}$; the relevant point is that h is strictly increasing and dominates f. We map f to $\tilde{f}$ defined by $\tilde{f}(0) = h(1)$ and $\tilde{f}(n+1) = h(\tilde{f}(n)+1)$. Note that $\tilde{f}$ is strictly increasing, and that $\tilde{f}(n) \geqslant f(n+1)$.

On the solution side, we map a function g, which we assume is strictly increasing, to the "step function" $\tilde{g}$ which on the interval $(g^{(n)}(0), g^{(n+1)}(0)]$ returns $g^{(n+1)}(0)$. So for $k = g^{(n+1)}(0)$ we have $\tilde{g}(k) = k < f(k)$, and so $\tilde{g}$ does not dominate f.

To show $\tilde{g}$ escapes f, we show that if $\tilde{g}$ is dominated by f, then g is dominated by $\tilde{f}$. For typographical clarity, let $a_n = g^{(n)}(0)$. Suppose that $\tilde{g}(k) \leqslant f(k)$ for all $k > a_m$. In particular, for $n \geqslant m$,

$$g(a_n) = a_{n+1} = \tilde{g}(a_n+1) \leqslant f(a_n+1) \leqslant \tilde{f}(a_n).$$

Now suppose that $n \geqslant m$ and $k \in (a_n, a_{n+1})$. Then

$$\begin{aligned} g(k) \leqslant g(a_{n+1}) = a_{n+2} = \tilde{g}(a_{n+1}+1) \leqslant f(a_{n+1}+1) \leqslant \\ h(a_{n+1}+1) = h(g(a_n)+1) \leqslant h(\tilde{f}(a_n)+1) = \tilde{f}(a_n+1) \leqslant \tilde{f}(k), \end{aligned}$$

using the fact that $g(a_n) \leqslant \tilde{f}(a_n)$ and that $\tilde{f}$ and h are increasing. □

Remark 7.18. The proof can be modifed to produce a strictly increasing $\tilde{g}$ which does not dominate $n \mapsto n^2$.

On our way, we define a strong variant of covering by a null set. For a set $A \subset \omega$ and a clopen set $C \subseteq 2^\omega$, we say that the *support* of C is contained in A ($\text{supp}(C) \subseteq A$) if membership in C is determined by examining the locations in A. Formally, C is the union of clopen sets $[p]$ where $p \in 2^A$ and $[p] = \{x \in 2^\omega : p \prec x\}$, where we extend the symbol $\prec$ to indicate extension of functions, not necessarily initial segment extension. We define the Weihrauch problem

- $\texttt{IndepCover}$: an instance is a closed, null set $P \subset 2^\omega$; a solution is a partition $\langle I_n \rangle$ of ω into intervals and a sequence $\langle C_n \rangle$ of clopen sets such that $\lambda(C_n) \leqslant 2^{-n}$, $\text{supp}(C_n) \subseteq I_n$ and for infinitely many n, $P \subseteq C_n$.

Lemma 7.19 (Bartozysnki,Shelah [5]). $\texttt{IndepCover} \to \texttt{Supset}(\mathscr{E}, \mathscr{N}) \star \texttt{Esc}$.

Proof. By lemma 7.17, we reduce $\texttt{IndepCover}$ to $\texttt{Supset}(\mathscr{E}, \mathscr{N}) \star \texttt{Esc}\neg\texttt{Dom}$.

Recall the notation $\sigma \rightharpoonup x$ and $\sigma \rightharpoonup X$ (see the proof of proposition 4.9). For a set P let $P^* = \bigcup_{\sigma \in 2^{<\omega}} \sigma \rightharpoonup P$. If P is closed null then P^* is null $\mathbf{\Sigma}^0_2$. We map such a set P to the pair (P^*, F), where $F \colon \mathscr{N} \to \omega^\omega$ is defined as follows. We are given a null set $V = \bigcap V_m$, and we assume that $P^* \subseteq V$. For $n < \omega$ let $P^n = \bigcup_{\sigma \in 2^n} \sigma \rightharpoonup P$. It is closed and null. First define $h(n,m)$ to be the least $s > n$ such that $P^n_s \subseteq V_{m,s}$.[12] Then define inductively $f(0) = 0$ and $f(n+1) = h(f(n), f(n))$. We define $F(V) = f$.

For a null set $V = \bigcap V_m$, and $m < s < \omega$ we let

$$Q_{m,s} = \bigcup \left\{ [p] \,:\, p \in 2^{[m,s)} \;\;\&\;\; [p] \subseteq V_{m,s} \right\}.$$

This is clopen, $\lambda(Q_{m,s}) \leqslant \lambda(V_m) \leqslant 2^{-m}$ and $\text{supp}(Q_{m,s}) \subseteq [m,s)$.

[12]Recall that the granularity of P^n_s and $V_{m,s}$ is at most s.

On the solution side, we are given a null set $V = \bigcap V_m$ and a non-decreasing function $g \in \omega^\omega$ such that $g(n) \geqslant n$. Define $I_n = (g(n), g(n+1)]$ and

$$C_n = \bigcup_{k \in I_n} Q_{k,g(n+1)}.$$

Then $\operatorname{supp}(C_n) \subseteq I_n$ and

$$\lambda(C_n) \leqslant \sum_{k \in I_n} \lambda(Q_{k,g(n+1)}) \leqslant \sum_{k \in I_n} 2^{-k} < 2^{-g(n)} \leqslant 2^{-n}.$$

We map (V, g) to $\langle I_n, C_n \rangle$.

To show that this works, suppose that $P^* \subseteq V$, let $f = F(V)$ (where P maps to (P^*, F)), and suppose that g is non-decreasing, escapes f but does not dominate f. Then there are infinitely many n for which

$$g(n) < f(n) < f(n+1) \leqslant g(n+1).$$

Fix such n; we claim that $P \subseteq C_n$. Now $P^{f(n)}_{f(n+1)} \subseteq V_{f(n),f(n+1)}$. We claim that $P \subseteq Q_{f(n),f(n+1)}$. Indeed, $P_{f(n+1)} \subseteq Q_{f(n),f(n+1)}$. For let $\rho \in 2^{f(n+1)}$ such that $[\rho] \subseteq P_{f(n+1)}$. Let $p = \rho\upharpoonright[f(n), f(n+1))$. For all $\sigma \in 2^{f(n)}$, $[\sigma\hat{\ } p] = [\sigma \rightharpoondown \rho] \subseteq V_{f(n),f(n+1)}$. Thus $[p] \subseteq V_{f(n),f(n+1)}$ and so $[p] \subseteq Q_{f(n),f(n+1)}$.

We then observe that as $f(n+1) \leqslant g(n+1)$, $Q_{f(n),f(n+1)} \subseteq Q_{f(n),g(n+1)}$; this is because $V_{f(n),f(n+1)} \subseteq V_{f(n),g(n+1)}$. Since $g(n) < f(n)$, we see that $Q_{f(n),g(n+1)} \subseteq C_n$. □

Remark 7.20. In lemma 7.17, by increasing values, we could require the range of g, and hence of $\tilde{g}$, to be a subset of any given infinite subset X of ω. That is, $\texttt{Esc} \leftrightarrow \texttt{Esc}\neg\texttt{Dom}_{\texttt{Coarse}}$, where $\texttt{Esc}\neg\texttt{Dom}_{\texttt{Coarse}}$ is the Weihrauch problem whose instances are pairs (f, X) where $f \in \omega^\omega$, $f > \mathrm{id}$, and $X \subseteq \omega$ is infinite; a solution is a non-decreasing function g which escapes f, does not dominate f, and $\operatorname{range} g \subseteq X$.

Applying this to the proof of the previous proposition, we see that

$$\texttt{IndepCover}_{\texttt{Coarse}} \to \texttt{Supset}(\mathscr{E}, \mathscr{N}) \star \texttt{Esc},$$

where $\texttt{IndepCover}_{\texttt{Coarse}}$ is the problem whose instances are pairs (P, X) where P is closed null and $X \subseteq \omega$ is infinite, and the solutions consist of a partition $\langle I_n \rangle$ of ω into intervals and a sequence $\langle C_n \rangle$ of clopen sets such that $\lambda(C_n) \leqslant 2^{-n}$, $\operatorname{supp}(C_n) \subseteq I_n$, for infinitely many n, $P \subseteq C_n$, and the endpoints of the intervals I_n are in X.

Another way to say this is that the instances are pairs $(P, \langle \tilde{I}_n \rangle)$ where the latter is a partition of ω into intervals, and for a solution $\langle I_n, C_n \rangle$, $\langle I_n \rangle$ is coarser than $\langle \tilde{I}_n \rangle$: every I_k is a union of $\tilde{I}_n$'s.

We need one last Weihrauch problem:

- $\texttt{GapEsc}$: an instance is a function $f \in \omega^\omega$; a solution is a non-decreasing function $g \in \omega^\omega$ such that for infinitely many n, $g(n+1) > f(g(n))$.

Lemma 7.21. $\texttt{Esc} \leftrightarrow \texttt{GapEsc}$.

Proof. To reduce $\texttt{Esc}$ to $\texttt{GapEsc}$, we map an instance f to itself. We map a solution g to a function h satisfying $h(g(n)) \geqslant g(n+1)$ for all n.

In the other direction, naively, it would seem that we could only reduce $\texttt{GapEsc} + \texttt{GapEsc}$ to $\texttt{Esc}$; there are two cases, depending on whether the escaping function also dominates or not. We can eliminate the dominating case using lemma 7.17:

we reduce $\texttt{GapEsc}$ to $\texttt{Esc}\neg\texttt{Dom}$. On the instance side, we map a function f, which we may assume is increasing, to $\tilde{f}(n) = f^{(n)}(0)$. On the solution side we use the identity map. Suppose that $\tilde{f}\,\texttt{Esc}\neg\texttt{Dom}\,g$. For infinitely many n, $g(n) < \tilde{f}(n) < \tilde{f}(n+1) < g(n+1)$. Then

$$g(n+1) > \tilde{f}(n+1) = f(\tilde{f}(n)) > f(g(n)). \qquad \square$$

Lemma 7.22. There is a computable function $U\colon \omega \to \omega$ such that for every sequence $\langle I_k\rangle_{k\leqslant m}$ of consecutive (nonempty) intervals, for every clopen set C such that $\lambda(C) \leqslant 1/2$, there are sets $S_k \subseteq I_k$ (computably obtained from $\langle I_k\rangle$ and C) such that $|S_k| \leqslant U(\min I_k)$ with the property that for every partial choice function $c\colon [0,m] \to I = \bigcup_{k\leqslant m} I_k$ such that $c(k) \in I_k$ for all $k \in \operatorname{dom} c$, if

$$E_c = \{x \in 2^\omega \,:\, (\forall k \in \operatorname{dom} c)\ x(c(k)) = 0\} \subseteq C,$$

then there is some $k \in \operatorname{dom} c$ such that $c(k) \in S_k$.

The point is that $|S_k|$ only depends on $\min I_k$, while $|I_k|$ may be large.

Proof. Fix a computable decreasing sequence $\alpha_0 > \alpha_1 > \alpha_2 > \cdots > 1$ of rational numbers such that $\prod_k \alpha_k < 2$. We then let

$$U(m) = 2^m \cdot \left\lceil -\log_2(1-\alpha_m^{-1})\right\rceil.$$

Suppose that $\langle I_k\rangle$ and C are given as in the lemma. We may assume that $\min I_0 = 0$, as U is increasing. Let $k \leqslant m$; let $n_k = \min I_k$. Let $\sigma \in 2^{n_k}$ such that $C \cap [\sigma] \neq \varnothing$. We then let

$$S_{k,\sigma} = \left\{\ell \in I_k \,:\, \forall q \in 2^{I_k}\left(q(\ell) = 0 \ \rightarrow\ \lambda(C|\sigma\hat{\ }q) > \alpha_k\lambda(C|\sigma)\right)\right\},$$

and let

$$S_k = \bigcup\left\{S_{\sigma,k} \,:\, \sigma \in 2^{n_k} \ \&\ C\cap[\sigma] \neq \varnothing\right\}.$$

We first verify that for each σ, $|S_{k,\sigma}| \leqslant -\log_2(1-\alpha_k^{-1})$. It then follows that $|S_k| \leqslant U(n_k)$, as $n_k \geqslant k$ and so $\alpha_{n_k} \leqslant \alpha_k$. Fix σ such that $C\cap[\sigma] \neq \varnothing$; since C is clopen, $\lambda(C|\sigma) > 0$. For brevity, in this argument let $S = S_{k,\sigma}$. For every $\ell < \omega$ let $D_\ell = \{x \in 2^\omega \,:\, x(\ell) = 0\}$. Note that $\lambda(\bigcup_{\ell\in S} D_\ell) = 1 - 2^{-|S|}$. It follows that there are at least $(1-2^{-|S|})2^{|I_k|}$ many $q \in 2^{I_k}$ such that $\lambda(C|\sigma\hat{\ }q) > \alpha_k\lambda(C|\sigma)$, that is, $\lambda(C\cap[q]|\sigma) \geqslant 2^{-|I_k|}\alpha_k\lambda(C|\sigma)$. Taking the union of $C\cap[q]$ for all such q, we see that

$$\lambda(C|\sigma) \geqslant (1-2^{-|S|})\cdot 2^{|I_k|}\cdot 2^{-|I_k|}\cdot\alpha_k\cdot\lambda(C|\sigma)$$

and since $\lambda(C|\sigma) > 0$, we get $(1-2^{-|S|})\alpha_k \leqslant 1$, and so the desired bound on $|S|$.

Now we are given a partial choice function c as in the statement of the lemma. Suppose that for all $k \in \operatorname{dom} c$, $c(k) \notin S_k$. We show that $E_c \nsubseteq C$.

We define recursively a sequence of strings $\sigma_k \in 2^{n_k}$, with the property that

$$\lambda(C|\sigma_k) \leqslant \frac{1}{2}\prod_{i<k}\alpha_i.$$

We start of course with $\sigma_0 = \langle\rangle$, with the measure bound given by the assumption that $\lambda(C) \leqslant 1/2$. Given σ_k for some $k \leqslant m$, there are three cases. If $k \notin \operatorname{dom} c$ then we let q_k be any $q \in 2^{I_k}$ satisfying $\lambda(C|\sigma_k\hat{\ }q) \leqslant \lambda(C|\sigma_k)$. Otherwise, if $C\cap[\sigma_k] = \varnothing$ then we let q_k be any $q \in 2^{I_k}$ such that $q(c(k)) = 0$. Otherwise, since $c(k) \notin S_{k,\sigma_k}$, and $C\cap[\sigma_k] \neq \varnothing$, we choose q_k to be some $q \in 2^{I_k}$ such that $q(c(k)) = 0$ and $\lambda(C|\sigma_k\hat{\ }q_k) \leqslant \alpha_k\lambda(C|\sigma_k)$. In all three cases we let $\sigma_{k+1} = \sigma_k\hat{\ }q_k$, and the measure bound holds. Then $\lambda(C|\sigma_{m+1}) \leqslant (1/2)\prod_{i\leqslant m}\alpha_i < 1$, so $[\sigma_{m+1}]$ witnesses that $E_c \nsubseteq C$ as required. ☐

Finally, the following proposition finishes the proof of $\texttt{IOE} \dashrightarrow \texttt{Supset}(\mathscr{E},\mathscr{N})$; with our other results, we get

$$\texttt{IOE} \to \texttt{Esc} \star \texttt{IndepCover}_{\texttt{Coarse}} \to \texttt{Esc} \star \texttt{Supset}(\mathscr{E},\mathscr{N}) \star \texttt{Esc} \to \texttt{Supset}(\mathscr{E},\mathscr{N})^{\star 3}.$$

Proposition 7.23. $\texttt{IOE} \to \texttt{Esc} \star \texttt{IndepCover}_{\texttt{Coarse}}$.

Proof. We show that $\texttt{IOE} \to \texttt{GapEsc} \star \texttt{IndepCover}_{\texttt{Coarse}}$. We use the computable function U given by lemma 7.22. Let $h(m) = m + m^{U(m)}$. Given a function $f \in \omega^\omega$ define $b_f\colon \omega \to \omega$ by letting $b_f(m) = \max\operatorname{range} f{\upharpoonright}(h^{(m+2)}(m))$.

Given a non-decreasing function g, we assume that $g(0) \geqslant 2$, and define intervals $J_k = [\ell_k, \ell_{k+1})$ and $I_k = [m_k, m_{k+1})$ (both partitioning ω) as follows: $|J_k| = U(m_k)$; $|I_k| = g(k)^{|J_k|}$. That is, $m_0 = \ell_0 = 0$; $\ell_{k+1} = \ell_k + U(m_k)$ and $m_{k+1} = m_k + g(k)^{U(m_k)}$. Note that $\ell_k \leqslant m_k$. Now by induction on k we can see that $m_{k+1} \leqslant h^{(k+1)}(g(k))$, and so that $\ell_{k+2} \leqslant h^{(k+2)}(g(k))$. It follows that if $g(n+1) > b_f(g(n))$ then $\operatorname{range} f{\upharpoonright}J_{n+1} \subseteq g(n+1)$. Thus, if $b_f \,\texttt{GapEsc}\, g$, then for infinitely many k we have $f{\upharpoonright}J_k \in g(k)^{J_k}$. Identifying I_n with $g(n)^{J_n}$, and writing $\mathbf{f}(n) = f{\upharpoonright}J_n$, we see that for infinitely many n, $\mathbf{f}(n) \in I_n$. It follows that $P^f = \{x \in 2^\omega : (\forall n)\ \mathbf{f}(n) \in I_n \ \to\ x(\mathbf{f}(n)) = 0\}$ is null.

Thus, on the solution side, we map $f \in \omega^\omega$ to the pair (b_f, F), where $F(g) = (P^f, \langle I_n\rangle)$ when $b_f \,\texttt{GapEsc}\, g$.

Keeping with g and $\langle I_n\rangle$, suppose that $\langle K_n, C_n\rangle \in \texttt{IndepCover}_{\texttt{sol}}$ and that $\langle K_n\rangle$ is coarser than $\langle I_n\rangle$. Fix $m > 0$; say $K_m = I_a \cup I_{a+1} \cup \cdots \cup I_b$. We define the sets $S_k \subseteq I_k$ for $k \in [a,b]$ as given by lemma 7.22 where we take $C = C_m$. Again thinking of I_k as $g(k)^{J_k}$, we think of every element of S_k as a function from J_k to $g(k)$. As $|S_k| \leqslant U(\min I_k) = U(m_k) = |J_k|$, we see that we can find a function $p_k \in g(k)^{J_k}$ which agrees with each element of S_k on some input.

On the solution side, we map $(g, \langle K_n, C_n\rangle)$ to the concatenation of the p_k's, that is to $p \in \omega^\omega$ defined by $p{\upharpoonright}J_k = p_k$.

To see that this works, suppose that indeed $b_f \,\texttt{GapEsc}\, g$ and, using the notation above, that $\langle K_m, C_m\rangle$ solves the problem $(P^f, \langle I_n\rangle)$. For infinitely many m, $P^f \subset C_m$. Define c to be the partial function which is the restriction of $\mathbf{f}$ to $\{k : \mathbf{f}(k) \in I_k\}$. For m such that $P^f \subset C_m$, if $K_m = I_a \cup \cdots \cup I_b$, let $c_m = c{\upharpoonright}[a,b]$. The set E_{c_m} is the collection of $q \in 2^{K_m}$ which are compatible with elements of P^f; since $\operatorname{supp}(C_m) \subseteq K_m$, we see that $E_{c_m} \subseteq C_m$. It follows that there is some $k \in [a,b]$ such that $\mathbf{f}(k) \in S_k$, and so that $f{\upharpoonright}J_k$ agrees with p_k on some input. As this happens for infinitely many m, we see that $f \,\texttt{IOE}\, p$, as required. ☐

8. Forcing

The most common way to prove the consistency of a strict inequality

$$\operatorname{Card}(A) < \operatorname{Card}(B)$$

is to start with a model of CH, and iterate (ω_2 many steps) a notion of forcing $\mathbb{P}$ that adds a real in $\mathrm{NL}^V(B)$ but no real in $\mathrm{NL}^V(A)$. That is, it adds some $b \in B_{\texttt{inst}}$ which is solved by no $\hat{b} \in V$; but every $a \in A_{\texttt{inst}} \cap V^{\mathbb{P}}$ is solved by some $\hat{a} \in V$. The standard argument is as follows. Let $\langle \mathbb{P}_\alpha\rangle$ be the iteration. In $V^{\mathbb{P}_{\omega_2}}$, $\operatorname{Card}(A) = \aleph_1$, because $V \cap A_{\texttt{sol}}$ solves all A-instances in $A_{\texttt{inst}} \cap V^{\mathbb{P}_{\omega_2}}$; no elements of $\mathrm{NL}^V(A)$

were added during the iteration. On the other hand, in $V^{\mathbb{P}_{\omega_2}}$, $\mathrm{Card}(B) = \aleph_2$. For suppose that $F \subset B_{\mathtt{sol}}$, $F \in V_{\mathbb{P}_{\omega_2}}$, and $|F| < \aleph_2$. Then there will be some $\alpha < \omega_2$ such that $F \subseteq V^{\mathbb{P}_\alpha}$. At step $\alpha+1$, then, we add a B-instance which has no solution in F.

Remark 8.1. There are several reasons for using represented spaces (definition 3.1). Here we see another one: the interpretation of a name may change between different models of set theory. Take for example the collection of names $\mathscr{M}$ for $\mathbf{\Sigma}^0_2$ meagre sets. If $V \subset W$ are transitive models of set theory, then $\mathscr{M}$ is absolute: $\mathscr{M}^V = \mathscr{M}^W \cap V$. However, likely for most $y \in \mathscr{M}$, the meagre set $M^W(y)$ named by y in W is strictly larger than the meagre set $M^V(y)$ named by y in V. When thinking of a problem such as $\mathtt{Pass}(\mathscr{M})$, we really mean the problem induced on the names rather than the meagre sets themselves: a $\mathtt{Pass}(\mathscr{M})$ solution (in W) for all instances in V (a Cohen generic over V) is a $y \in \omega^\omega$ such that for all $x \in \mathscr{M}^V$, $y \notin M^W(x)$, rather than $y \notin M^V(x)$ (which trivially holds for all $y \in W \backslash V$). It is also important that relations such as $\mathtt{Pass}(\mathscr{M})$ and $\mathtt{Supset}(\mathscr{M})$ are absolute.

The relationship between computability and set theory here is imprecise. In many cases, the notion of forcing $\mathbb{P}$ itself is a represented space, and we can force with the computable elements of $\mathbb{P}$, and hope to get a real in $\mathrm{NL}(B) \backslash \mathrm{NL}(A)$. In these cases, the failure of the implication $B \to A$ is witnessed at the computable level as well.

8.1. **Basic examples.** Cohen forcing is unusual in that it is the unique countable notion of forcing; all conditions are computable. A Cohen real does not make the collection of reals in the ground model meagre. And indeed, the argument is effective.

Lemma 8.2 (Rupprecht [41]). A sufficiently Cohen generic real is not weakly meagre engulfing.

So a Cohen generic gives an oracle in $\mathrm{NL}(\mathtt{Capture}(\mathscr{M})) \backslash \mathrm{NL}(\mathtt{Pass}(\mathscr{M}))$, showing that $\mathtt{Capture}(\mathscr{M}) \nrightarrow \mathtt{Pass}(\mathscr{M})$, effectively.

Proof. Suppose that p is a condition that forces that $\langle T_n \rangle$ is a sequence of uniformly partial computable functions from 2^ω to 2^ω, and that for each n, $T_n(g)$ is a nowhere dense tree. Thus, for each n and $\sigma \in 2^{<\omega}$, densely below p we can find conditions q and extensions $\tau \succcurlyeq \sigma$ such that $T_n(q)$ declares that $\tau \notin T_n(g)$. Further, such pairs (q, τ) can be found effectively. We can thus construct a computable point $x \in 2^\omega$ by initial segments; at step (r, n), for some condition r extending p, we find q extending r and τ extending the initial segment of x we have so far such that $\tau \notin T_n(r)$. □

Remark 8.3. The argument above is identical to the argument showing that the ground model reals in a Cohen extension are not meagre. In that argument we need to first prove a *continuous reading of names*, which tells us that every real in the extension is the image of the generic by a continuous function with code in the ground model. On the other hand, unlike the computability proof, in set theory we don't need to worry about how effective is the search for (q, τ).

Proving that an ω_2-iteration of Cohen over a model of CH gives a model of $\mathbf{non}(\mathscr{M}) < \mathbf{cov}(\mathscr{M})$ requires an *iteration theorem*, which states that reals in $\mathrm{NL}^V(\mathtt{Pass}(\mathscr{M}))$ are not added at limit steps of the iteration. In the Cohen case this is not too difficult, but with notions of forcing that are proper rather than c.c.c., such an iteration theorem may take some work. As we see, the corresponding argument in computability theory does not require such a result.

Figure 4 gives the dividing line for Cohen forcing in the Cichoń diagram.

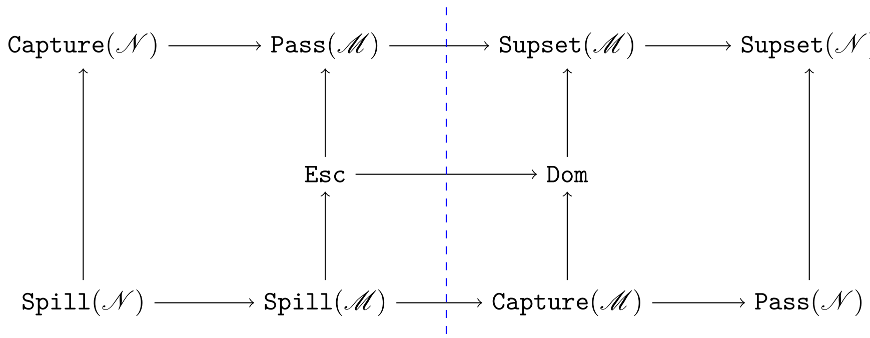

FIGURE 4. Cohen forcing and the Cichoń diagram. Cohen forcing adds a real in $\mathrm{NL}(A)$ for all problems A on the right of the dividing line, but not for problems on its left. Similarly, an iteration of Cohen forcing will make the associated cardinals on the right large, and keep the cardinals on the left small.

Remark 8.4. In the Cohen extension, the ground model reals are null. This is an immediate corollary of the morphism $\texttt{Capture}(\mathscr{M}) \to \texttt{Pass}(\mathscr{N})$, as it gives $\mathrm{NL}^V(\texttt{Capture}(\mathscr{M})) \subseteq \mathrm{NL}^V(\texttt{Pass}(\mathscr{N}))$.

The next simplest notion of forcing is perhaps random real forcing, one version of which is forcing with closed sets of positive measure. The conditions with computable names are the Π^0_1 classes with positive measure. Forcing with these does not give a generic for full random forcing; a sufficiently generic real will be weakly 2-random, but not 2-random. We do get a Schnorr random, i.e., an oracle in $\mathrm{NL}(\texttt{Capture}(\mathscr{N}))$. A generic will also be computably dominated and will not be weakly null engulfing (Rupprecht [41, VI.35], based on the argument by Kunen [29] showing that in the random model, the ground model reals are not null.) This gives the diagram in fig. 5.

8.2. **Two step iterations.** Before we proceed, we make a side remark on two step iterations.

Lemma 8.5. Let A and B be absolute Weihrauch problems, and let $V \subseteq W \subseteq U$ be transitive models of set theory. If $x \in W \cap \mathrm{H}^V(A)$ and $y \in U \cap \mathrm{H}^W(B)$, then $(x, y) \in \mathrm{H}^V(A \star B)$.

Proof. Let (a, F) be an instance of $A \star B$. Then $a\,A\,x$. Since F is coded in V, $F(x) \in W$, and so $F(x)\,B\,y$. □

So from the morphism $\texttt{Pass}(\mathscr{M}) \to \texttt{IOE} \star \texttt{IOE}$ we see that if we add an $\texttt{IOE}$ function g relative to V, and then an $\texttt{IOE}$ function relative to $V[g]$, then in the second extension there must be a real Cohen over V.

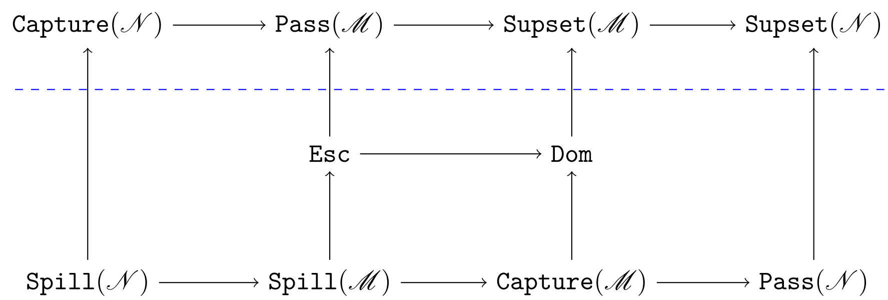

FIGURE 5. Random forcing and the Cichoń diagram; the forcing adds solutions (and increases cardinals) for problems above the dividing line.

Zapletal [48] showed that the iteration was necessary: he found a notion of forcing that adds an `IOE` function without adding a Cohen real. His notion of forcing shows that $\texttt{Pass}(\mathscr{M}) \nrightarrow \texttt{IOE}$, even when we consider non-computable definable morphisms, say Borel morphisms, for which the delicate argument of lemma 4.22 does not apply.

8.3. **Working relative to ideals.** Sometimes, however, computability does not reveal the full picture. The most familiar example is the morphism

$$\texttt{Capture}(\mathscr{M}) \to \texttt{Dom},$$

which cannot be reversed. However, $\mathrm{NL}(\texttt{Capture}(\mathscr{M})) = \mathrm{NL}(\texttt{Dom})$: in the unusual direction, every hyperimmune degree computes a weakly 1-generic. This relies on a “time trick”, special to computability. Indeed, in set theory, it is possible to get an extension with a real in $\mathrm{NL}^V(\texttt{Dom})$ but none in $\mathrm{NL}^V(\texttt{Capture}(\mathscr{M}))$, which after iterating gives the consistency of $\mathbf{cov}(\mathscr{M}) < \mathfrak{d}$. One of the most straightforward notions is *Miller forcing*, also called *rational perfect tree* forcing. The conditions are trees $T \subseteq \omega^{<\omega}$ with no dead ends, no isolated paths, and in which every split is an infinite split: if $\sigma \in T$ and there is more than one $k < \omega$ such that $\sigma\hat{\ }k \in T$, then there are infinitely many such k. Extension is as is usual with most tree forcings: subset.

Denote this notion of forcing by $\mathbb{P}$. On one hand, it is not difficult to see that $\mathbb{P}$ adds an escaping function: for every function f, for every condition $T \in \mathbb{P}$, we can refine the condtion by removing, for every splitting node $\tau \in T$, all extensions $\tau\hat{\ }k$ for $k \leqslant f(|\tau|)$.

On the other hand, Miller forcing has the *Laver tracing* property, and so does not add a Cohen generic. Namely, for every order function h, for every function $f \in \omega^\omega \cap V^{\mathbb{P}}$ which is bounded by some function in V, there is some h-Trace in V which traces f. Once one has continuous reading of names, we take a continuous function $\Phi \in V$ and a condition T and perform a kind of fusion argument. First, by thinning, we may assume that if τ is a n^{th} splitting node (it extends $k-1$ many splitting nodes) then for every immediate extension $\tau\hat{\ }k$ of τ on T, $\Phi(\tau\hat{\ }k, n)\downarrow$. Next, we thin one splitting node at a time to obtain a trace. For example, if τ is the shortest splitting node (the *stem* of the tree) then we can force a single value for $\Phi(g, 0)$ (g being the generic path): because $\Phi(g)$ is V-bounded, say by some function $\tilde{h} \in V$, we may assume that for every extension $\tau\hat{\ }k$ of τ, the value $\Phi(\tau\hat{\ }k, 0)$ is smaller than $\tilde{h}(0)$, in particular, only finitely many values appear; so one value appears infinitely often. We take one such value a_0 and remove all extensions $\tau\hat{\ }k$ for which $\Phi(\tau\hat{\ }k, 0) \neq a_0$. The 0^{th} element of our trace will be $\{a_0\}$.

Next, consider the next level. Let τ_k be the next splitting extension above $\tau\hat{\ }k$. For each k, some value a_k appears as $\Phi(\tau_k\hat{\ }l, 1)$ for infinitely many l. Among the various k, one value a appears infinitely often. We remove from T all $\tau\hat{\ }k$ such that $a_k \neq a$, *except possibly for the least such* k – we will be "protecting" more and more successors of τ, so that after ω-many steps, infinitely many successors will remain. The next element of the trace is $\{a, a_k\}$.

We see that the reason that this argument does not work computably is that even if T is computable, the construction of the thinned tree is not computable: it requires answers to the requests "give us a value that occurs infinitely often." This can be done with the Turing jump of T. This shows that if we force with all arithmetic conditions, we will obtain a function escaping all arithmetic functions, which computes no arithmetically generic real. Indeed, even with the help of any arithmetic oracle, no such generic can be computed. Recall from section 5 that we defined $\mathrm{NL}^r(A)$ for reducibilities $\leqslant_r$ implied by Turing. If I is an ideal of Turing degrees then the reducibility $\leqslant_I$ (Turing modulo I) is defined by $x \leqslant_I y$ if there is some $z \in I$ such that $x \leqslant_{\mathrm{T}} z \oplus y$. We write $\mathrm{NL}^I(A)$ for the corresponding non-lowness class. The argument above sketched the proof of the following:

Proposition 8.6. *If I is a jump ideal then* $\mathrm{NL}^I(\mathtt{Capture}(\mathscr{M})) \neq \mathrm{NL}^I(\mathtt{Dom})$.

We remark that in [24], Kihara showed that Miller forcing has a sufficiently effective continuous reading of names, so that this inequality holds for relative hyperarithmetic reducibility as well. In contrast with proposition 8.6, the argument that $\mathrm{NL}(\mathtt{Capture}(\mathscr{M})) = \mathrm{NL}(\mathtt{Dom})$ applies to any ideal I which has a maximal degree of lowness for weak genericity reducibility. By relativising lowness for weak genericity, we see that these are the ideals in which both the DNR principle and the ESC principle (also named the HI principle) fail. That is, ideals I which contain a real z relative to which in I there is no DNR function, nor a function escaping all z-computable functions. On the other hand, proposition 8.6 can be strengthened to apply to ideals I satisfying the $\mathrm{DNR}(x')$ principle: for every $x \in I$ there is $f \in I$ which is DNR relative to x', even if $x' \notin I$. The reason is that if we are given a value $\Phi(\tau\hat{\ }k, n)$ which does not occur cofinitely often, we can thin to ensure that the generic does not compute an IOE function, that is, that the generic is in $\mathrm{NL}^I(\mathtt{Dom})\backslash\mathrm{NL}^I(\mathtt{AEDiff})$; and recall that $\mathtt{Capture}(\mathscr{M}) \to \mathtt{AEDiff} \to \mathtt{Dom}$. The following question though remains open:

Question 8.7. *For which Turing ideals I is it the case that* $\mathrm{NL}^I(\mathtt{Capture}(\mathscr{M})) \neq \mathrm{NL}^I(\mathtt{Dom})$*?*

The same kind of question can be asked about the two other "collapses" in the computable Cichoń diagram:

$$\mathrm{NL}(\mathtt{Spill}(\mathscr{N})) = \mathrm{NL}(\mathtt{Spill}(\mathscr{M})) = \mathrm{NL}(\mathtt{Esc}),$$

namely: the high degrees coincide with the meagre engulfing and null engulfing degrees. Unlike the previous example, here it is known that the coincidences hold for relative hyperarithmetic reducibility. This is not the same though as Turing modulo Δ^1_1, for which the question is still open. Hechler forcing and Laver forcing give separations in set theory. The associated constructions can be pushed in ideals which are models of arithmetic transfinite recursion ($\mathtt{ATR}_0$). Not much is known otherwise.

References

[1] Uri Andrews, Mingzhong Cai, David Diamondstone, Carl Jockusch, and Steffen Lempp. Asymptotic density, computable traceability, and 1-randomness. *Fund. Math.*, 234(1):41–53, 2016.

[2] Tomek Bartoszyński. Additivity of measure implies additivity of category. *Trans. Amer. Math. Soc.*, 281(1):209–213, 1984.

[3] Tomek Bartoszyński. Combinatorial aspects of measure and category. *Fund. Math.*, 127(3):225–239, 1987.

[4] Tomek Bartoszyński and Haim Judah. *Set theory: on the structure of the real line*. A K Peters Ltd., Wellesley, MA, 1995.

[5] Tomek Bartoszyński and Saharon Shelah. Closed measure zero sets. *Ann. Pure Appl. Logic*, 58(2):93–110, 1992.

[6] Andreas Blass. Reductions between cardinal characteristics of the continuum. In *Set theory (Boise, ID, 1992–1994)*, volume 192 of *Contemp. Math.*, pages 31–49. Amer. Math. Soc., Providence, RI, 1996.

[7] Andreas Blass. Combinatorial cardinal characteristics of the continuum. In *Handbook of set theory. Vols. 1, 2, 3*, pages 395–489. Springer, Dordrecht, 2010.

[8] Vasco Brattka. Computability and analysis, a historical approach. In *Pursuit of the universal*, volume 9709 of *Lecture Notes in Comput. Sci.*, pages 45–57. Springer, [Cham], 2016.

[9] Vasco Brattka, Guido Gherardi, and Alberto Marcone. The Bolzano-Weierstrass theorem is the jump of weak König's lemma. *Ann. Pure Appl. Logic*, 163(6):623–655, 2012.

[10] Jörg Brendle, Andrew Brooke-Taylor, Keng Meng Ng, and André Nies. An analogy between cardinal characteristics and highness properties of oracles. In *Proceedings of the 13th Asian Logic Conference*, pages 1–28. World Sci. Publ., Hackensack, NJ, 2015.

[11] Jörg Brendle and André Nies. Up to $2 \cdot 2 \cdot 2^{\aleph_0}$ cardinal invariants, and their counterparts in computability theory. In *Logic Blog 2015*. Available at http://arxiv.org/abs/1602. 04432, 2015.

[12] Chi Tat Chong, André Nies, and Liang Yu. Lowness of higher randomness notions. *Israel J. Math.*, 166:39–60, 2008.

[13] Samuel Coskey, Tamás Mátrai, and Juris Steprāns. Borel Tukey morphisms and combinatorial cardinal invariants of the continuum. *Fund. Math.*, 223(1):29–48, 2013.

[14] François G. Dorais, Damir D. Dzhafarov, Jeffry L. Hirst, Joseph R. Mileti, and Paul Shafer. On uniform relationships between combinatorial problems. *Trans. Amer. Math. Soc.*, 368(2):1321–1359, 2016.

[15] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic randomness and complexity*. Theory and Applications of Computability. Springer, New York, 2010.

[16] D. H. Fremlin. Real-valued-measurable cardinals. In *Set theory of the reals (Ramat Gan, 1991)*, volume 6 of *Israel Math. Conf. Proc.*, pages 151–304. Bar-Ilan Univ., Ramat Gan, 1993.

[17] Guido Gherardi and Alberto Marcone. How incomputable is the separable Hahn-Banach theorem? *Notre Dame J. Form. Log.*, 50(4):393–425 (2010), 2009.

[18] Martin Goldstern and Saharon Shelah. Many simple cardinal invariants. *Arch. Math. Logic*, 32(3):203–221, 1993.

[19] Noam Greenberg and Joseph S. Miller. Lowness for Kurtz randomness. *J. Symbolic Logic*, 74(2):665–678, 2009.

[20] Noam Greenberg and Benoit Monin. Higher randomness and genericity. *Forum Math. Sigma*, 5:e31, 41, 2017.

[21] Noam Greenberg and Dan Turetsky. Strong jump-traceability. Bulletin of Symbolic Logic, to appear.

[22] Denis R. Hirschfeldt, Carl G. Jockusch, Jr., Timothy H. McNicholl, and Paul E. Schupp. Asymptotic density and the coarse computability bound. *Computability*, 5(1):13–27, 2016.

[23] Jakob Kellner. Even more simple cardinal invariants. *Arch. Math. Logic*, 47(5):503–515, 2008.

[24] Takayuki Kihara. Higher randomness and lim-sup forcing within and beyond hyperarithmetic. In *Sets and computations*, volume 33 of *Lect. Notes Ser. Inst. Math. Sci. Natl. Univ. Singap.*, pages 117–155. World Sci. Publ., Hackensack, NJ, 2018.
[25] Bjø rn Kjos-Hanssen, André Nies, Frank Stephan, and Liang Yu. Higher Kurtz randomness. *Ann. Pure Appl. Logic*, 161(10):1280–1290, 2010.
[26] Bjø rn Kjos-Hanssen, Frank Stephan, and Sebastiaan A. Terwijn. Covering the recursive sets. *Ann. Pure Appl. Logic*, 168(4):804–823, 2017.
[27] Bjørn Kjos-Hanssen, Wolfgang Merkle, and Frank Stephan. Kolmogorov complexity and the recursion theorem. In *STACS 2006*, volume 3884 of *Lecture Notes in Comput. Sci.*, pages 149–161. Springer, Berlin, 2006.
[28] Bjørn Kjos-Hanssen, André Nies, and Frank Stephan. Lowness for the class of Schnorr random reals. *SIAM J. Comput.*, 35(3):647–657 (electronic), 2005.
[29] Kenneth Kunen. Random and Cohen reals. In *Handbook of set-theoretic topology*, pages 887–911. North-Holland, Amsterdam, 1984.
[30] Donald A. Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Z. Math. Logik Grundlagen Math.*, 12:295–310, 1966.
[31] Arnold W. Miller. Some properties of measure and category. *Trans. Amer. Math. Soc.*, 266(1):93–114, 1981.
[32] Kenshi Miyabe. Truth-table Schnorr randomness and truth-table reducible randomness. *MLQ Math. Log. Q.*, 57(3):323–338, 2011.
[33] Benoit Monin. An answer to the gamma question. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 730–738. ACM, 2018.
[34] Benoit Monin and André Nies. Muchnik degrees and cardinal characteristics. arXiv:1712.00864.
[35] Benoit Monin and André Nies. A unifying approach to the Gamma question. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2015)*, pages 585–596. IEEE Computer Soc., Los Alamitos, CA, 2015.
[36] André Nies. *Computability and randomness*, volume 51 of *Oxford Logic Guides*. Oxford University Press, Oxford, 2009.
[37] André Nies. Lowness, randomness, and computable analysis. In Adam Day et al., editor, *Computability and complexity: Essays dedicated to Rodney G. Downey on the occasion of his 60th birthday.*, volume 10010 of *Lecture Notes in Comput. Sci.*, pages 738–754. Springer, 2017.
[38] Noboru Osuga and Shizuo Kamo. Many different covering numbers of Yorioka's ideals. *Arch. Math. Logic*, 53(1-2):43–56, 2014.
[39] Janusz Pawlikowski and Ireneusz Recł aw. Parametrized Cichoń's diagram and small sets. *Fund. Math.*, 147(2):135–155, 1995.
[40] Jean Raisonnier and Jacques Stern. The strength of measurability hypotheses. *Israel J. Math.*, 50(4):337–349, 1985.
[41] Nicholas Rupprecht. *Effective Correspondents to Cardinal Characteristics in Cichoń's Diagram*. PhD thesis, University of Michigan, 2010.
[42] Nicholas Rupprecht. Relativized Schnorr tests with universal behavior. *Arch. Math. Logic*, 49(5):555–570, 2010.
[43] C.-P. Schnorr. A unified approach to the definition of random sequences. *Math. Systems Theory*, 5:246–258, 1971.
[44] Frank Stephan and Liang Yu. Lowness for weakly 1-generic and Kurtz-random. In *Theory and applications of models of computation*, volume 3959 of *Lecture Notes in Comput. Sci.*, pages 756–764. Springer, Berlin, 2006.
[45] Sebastiaan A. Terwijn and Domenico Zambella. Computational randomness and lowness. *J. Symbolic Logic*, 66(3):1199–1205, 2001.
[46] Peter Vojtáš. Topological cardinal invariants and the Galois-Tukey category. In *Recent developments of general topology and its applications (Berlin, 1992)*, volume 67 of *Math. Res.*, pages 309–314. Akademie-Verlag, Berlin, 1992.
[47] Klaus Weihrauch. *Computable analysis*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2000. An introduction.
[48] Jindřich Zapletal. Dimension theory and forcing. *Topology Appl.*, 167:31–35, 2014.

Department of Mathematics, Victoria University of Wellington, Wellington, New Zealand
E-mail address: greenberg@msor.vuw.ac.nz
URL: http://homepages.mcs.vuw.ac.nz/~greenberg/

Department of Mathematics, Victoria University of Wellington, Wellington, New Zealand

Department of Mathematics, Victoria University of Wellington, Wellington, New Zealand
E-mail address: dan.turetsky@msor.vuw.ac.nz