A HIERARCHY OF COMPUTABLY ENUMERABLE DEGREES, II: SOME RECENT DEVELOPMENTS AND NEW DIRECTIONS

ROD DOWNEY, NOAM GREENBERG, AND ELLEN HAMMATT

Abstract. A transfinite hierarchy of Turing degrees of c.e. sets has been used to calibrate the dynamics of families of constructions in computability theory, and yields natural definability results. We review the main results of the area, and discuss splittings of c.e. degrees, and finding maximal degrees in upper cones.

1. Vaughan Jones

This paper is a submission for the Vaughan Jones memorial issue of the New Zealand Journal of Mathematics. Vaughan had a massive influence on World and New Zealand mathematics, both through his work and his personal commitment. Downey first met Vaughan in the early 1990's, around the time he won the Fields Medal. Vaughan had a vision of improving mathematics in New Zealand, and hoped his Fields Medal could be leveraged to this effect. It is fair to say that at the time, maths in New Zealand was a wee bit sleepy. It was fortunate that not only was there a lot of press coverage, at the same time another visionary was operating in New Zealand: Sir Ian Axford. Ian pioneered the Marsden Fund, and a group of mathematicians gathered by Vaughan, Marston Conder, David Gauld, Gaven Martin, and Rod Downey, put forth a proposal for summer meetings to the Marsden Fund. Based on the huge influence of Vaughan, and the fact that Ian chaired the committee, this was approved and this group set about bringing in external experts to enrich the NZ scene. It is hard for someone now to understand what little exposure there was at the time to research trends at the time. Vaughan would attend the meetings, guide them, help with the washing up, talk to the students, help with all academic things. He never lost his Kiwi commitment. The rest, as they say, is history, and New Zealand mathematics now full of excellent world class mathematicians.

On a personal note, Downey talked mathematics (and life) with Vaughan, had amazing "field trips" down to the Catlins, for example, where Vaughan entertained the Fortrose locals with kite boarding, and had this amazing nearly 30 year relationship. Vaughan had great vision for mathematics and saw great value, for example, in mathematical logic. Certainly he was a great mentor and friend to Downey. Greenberg would also like to acknowledge Vaughan's support over the last 15 years, and many fruitful interactions in NZMRI meetings, Auckland and Southland. His is a terrible loss.

The authors' research was supported by the Marsden Fund of New Zealand. One of the Theorems was part of the third author's MSc Thesis.

2. Introduction

The goal of this paper is to look at the hierarchy of Turing degrees introduced by Downey and Greenberg [16, 22]. Whilst we concentrate on recent work not covered in either the monograph [22] or the BSL survey [16], we will give enough background material for the reader to understand the issues addressed by the hierarchy. We will give some proofs and proof sketches so the the reader can understand the techniques involved.

2.1. **Background.**

This paper can be read by two audiences:

- A lay logician could read it to see some of the issues involved in a sub-area of mathematical logic.
- A worker in computability theory can read this in more detail to glean ideas about the development of this new classification hierarchy and its recent generalizations.

For the sake of the first audience, we will give this short background section. The second one can skip this.

The area of this work is modern *computability theory*. This is an area of mathematics devoted to understanding what part of mathematics can be performed by a machine. *Computational complexity theory* is then concerned with what resources are needed if it can. Computability theory is also concerned with calibrating how hard a problem is, if it cannot be computed. As we see below, both computability theory and complexity theory use various *hierarchies* and *reducibilities* as classification tools which allow us to understand how problems relate in terms of their algorithmic difficulty.

In the mid-1930's, Church, Kleene, Post and famously Turing [63] gave a mathematical definition of a the intuitive idea of a computable function, which we view as being one that could be emulated by a Turing machine, in accordance with the Church-Turing Thesis. Today, computers are ubiquitous, and we can consider a function as being computable if we can write code to compute the function. A key idea of these studies is that algorithms can be treated as data, and hence there is an algorithm which enumerates all algorithms (or all Turing Machines), and hence a computable list $\{\varphi_e \mid e \in \mathbb{N}\}$ of all partial computable functions.

Famously, Church and Turing solved the *Entscheidungsproblem*, Turing by showing that the algorithmically undecidable halting problem "Does the e^{th} Turing machine on input e halt?" could be coded into first order logic and hence first order logic was *undecidable*. We will use $\varnothing'$ to denote the collection of all halting algorithms. This is an example of a *computably enumerable* set (abbreviated *c.e.*); where A is computably enumerable iff it is empty or the range of a of computable function g. The idea is that we can consider $A = \{g(0), g(1), g(2), \dots\}$ as being enumerated one number at a time, but unlike decidable sets, this enumeration may not be in order.

Reducibilities calibrate the complexity of sets. $A \leqslant B$ informally means that A is no more complicated than B. The simplest reducibility is m-reducibility $A \leqslant_m B$, which is defined to mean that there is a computable function f such that for all x, $x \in A$ iff $f(x) \in B$. In classical mathematics, an example of such a reduction is that an $n \times n$ matrix is nonsingular iff it has non-zero determinant: the question of singularity reduces to the calculation of a number.

A generalization of m-reducibility is called truth-table reducibility, $A \leqslant_{tt} B$, defined as follows. On input x we generate a finite collection $\sigma(x)$ of queries about memebership and non-membership in B, such that $x \in A$ iff B satisfies all these queries. The most general reducibility is *Turing reducibility*, $A \leqslant_T B$, meaning that using B as read-only memory, there is a procedure which can determine $x \in A$ using a computable process generating a finite number of quieries to B. Unlike truth-table reducibility, these queries are adaptive in that they can depend on the result of pervious queries; and in general, the computation process may or may not halt. If $A \leqslant_T B$ then we say that A is B-computable. Following Turing, we sometimes refer to the set B as an "oracle" for a computation, or say that the computational process is *relativized* to B. The relation $\leqslant_T$ is reflexive and transitive (it is a pre-ordering), and thus determines an equivalence relation $A \equiv_T B$ (Turing *equivalence*) defined by $A \leqslant_T B$ and $B \leqslant_T A$; and an induced partial ordering of the equivalence classes, which are called *Turing degrees*. We can relativize the halting problem to an oracle B, to obtain B', the *Turing jump* of B (the collection of halting algorithms which have access to B); this induces an increasing function on the Turing degrees. We reamrk that polynomial time reducibility, which is the polynomial time version of Turing reducibility, implies truth table reducibility.

Ever since Post's seminal paper [57], two recurrent themes in computability theory have been understanding the *dynamic* nature of constructions, and *definability* in the natural structures of computability theory such as the computably enumerable sets (ordered by inclusion), and sub-orderings of the Turing degrees. For this paper, an important example of this phenomenom is an old result of Shoenfield called the *Limit Lemma*, which says that the sets computable from the halting problem are those which can be pointwise computed algorithmically with a finite number of mind changes.

Theorem 2.1 (Shoenfield [58])**.** A function $g\colon \mathbb{N} \to \mathbb{N}$ is computable relative to the halting problem $\varnothing'$ iff it has a *computable approximation*: a sequence $g_0, g_1, \ldots$ of uniformly computable functions such that for all n, $g(n) = \lim_s g_s(n)$, meaning that for all but finitely many s, $g(n) = g_s(n)$.

The result holds for sets $A \subseteq \mathbb{N}$ as well, by identifying a set with its characteristic function.

A beautiful and deep example of this definability/dynamic phenomenom is the definable solution to Post's problem of Harrington and Soare [37]. Post asked whether there was a computably enumerable set A with $\varnothing <_T A <_T \varnothing'$. The context was that up to the time of the solution, all undecidable c.e. sets were basically the halting set in disguise; certainly they were all Turing equivalent to $\varnothing'$. Post's problem was solved by Friedberg [35] and Muchnik [52] using an intricate combinatorial technique, which has become a hallmark of the area: *the priority method*. The disappointing aspect of the solution is that the c.e. set A is only the solution to a construction. It seems unnatural. There remain to this day many questions relating to whether there is a natural solution to Post's problem. The Harrington-Soare result shows that there is a formula of predicate logic, in the langauge of the partial ordering of c.e. sets under inclusion, such that some c.e. set satisfies the formula, and any c.e. set satisfying the formula must be of intermediate Turing degree.

2.2. **The present paper.** As we have said, along with [16], the goal of this paper is to report on the current results of a program introduced by Downey, Greenberg and some co-authors, which seeks to understand the fine structure of relationship between *dynamic* properties of sets and functions, their *definability*, and their *algorithmic complexity*. Many of the resuls and details can be found in Downey and Greenberg's monograph [24]. In that monograph, along with the companion papers [23] and [21], the first two authors introduce a new hierarchy of c.e. degrees (Turing degrees of c.e. sets) based on the complexity of approximations of functions in these degrees. Logic is full of classification hierarchies. The reader might well ask why we need yet another hierarchy in computability theory. As the first two authors claim in [16, 22], the new hierarchy gives insight into

(i) A new methodology for classifying and unifying the combinatorics of a number of constructions from the literature.
(ii) New natural definability results in the c.e. degrees. These definability results are in the low$_2$ degrees and hence are not covered by the metatheorems of Nies, Shore and Slaman [55]. Moreover they are amongst the very few natural definability results in the theory of the c.e. Turing degrees.
(iii) The introduction of a number of construction techniques which are injury-free and highly non-uniform.

3. The new hierarchy

There is a depressingly ad hoc aspect to a lot of classical computability theory. Many combinatorial constructions occur and have a similar flavour. The question is: are there any underlying principles which allow us to classify the combinatorics into classes? There have been historical examples.

Martin [47] was a poineer here. He showed that sets which resemble the halting set have a certain domination property: if $A' \equiv_T (\varnothing')'$ then A is called a high set[1] and it is one indistinguishable from the halting problem in terms of its jump. Prior to Martin's paper, there had been a number of constructions which had similar feel. Martin showed that A is high iff A computes a function g such that for all computable f, there is an n such that for all $m \geqslant n$, $g(m) > f(m)$. We say that g *dominates* f. Using this, Martin showed that a large collection of known classes of c.e. sets existed in exactly the high c.e. degrees: dense simple, maximal, hh-simple and other similar kinds of c.e. sets; see Soare [61]. Martin's characterization has proved extremely useful ever since. For example, using this it is possible to show sets of high degree are those with with maximal EX-degree in Gold-style learning [13] and high degrees are precisely those which allow for separation of Martin-Löf, Schnorr and Computable randomness [56].

In relativised form (see e.g. Lerman [46]), Martin's result allows for a characterization of the low$_2$ degrees, where A is low$_2$ iff $A'' \equiv_T \varnothing''$. That is,

Theorem 3.1 (Martin [47])**.** A degree $\mathbf{d} < \mathbf{0}'$ is nonlow$_2$ iff for every function $h \leqslant_T \varnothing'$, there is a $\mathbf{d}$-computable function g which is not dominated by h.

Again this characterization has been widely used, for example in Jockusch's proof that every non-low$_2$ degree bounds a 1-generic one. We use a function h which grows sufficiently quickly, so that for all strings σ of length x and all $e \leqslant x$, if σ has an extension in V_e, the e^{th} c.e. set of strings, then we can see such an

[1] Recall that X is called low if $X' \equiv_T \varnothing'$.

extension by stage $h(x)$. Then given a non-low$_2$ set D there is a function $g \leqslant_T D$ which infinitely often escapes h. We use $g(x)$ as our search space to figure out which requirement to pursue. Then a standard finite injury argument works. See, for example, Lerman [46], or Downey and Shore [17].

Another example is the class of promptly simple degrees (see Ambos-Spies, Jockusch, Shore and Soare [4]), which characterises a class of sets which resemble the halting set in terms of a certain immediacy of change. One recent example of current great interest is the class of K-trivial reals (see Downey, Hirschfeldt, Nies and Stephan [25], and Nies [54, 53]), which are known to coincide with many other "lowness" constructions. These are sets which resemble the computable sets in terms of initial segment complexity.

Our concern will be the classification of c.e. sets in terms of how hard it is to approximate the things they compute. To this end we will look at the fine structure of the Limit Lemma. The next section gives a key tool.

3.1. **α-c.a. functions.** Our main concern will be functions and sets which are Δ^0_2-definable, being the functions $g \leqslant_T \varnothing'$. As mentioned, the Limit Lemma (Theorem 2.1) states that these are the functions that have computable approximations $g = \lim_s g_s$. The idea is to classify their degrees according to the complexity of a bound on the "mind change" function $\#\{s : g_{s+1}(x) \neq g_s(x)\}$.

In [22], we follow a classification of Δ^0_2 functions defined by Ershov in [32, 33, 34]. A witness to the approximation $\langle g_s \rangle$ stabilising on an input is a counting down some ordinal α. The greater the ordinal α, the more opportunities we have for changes.

A notion of reducibility stronger than Turing and weaker than truth-table is called *weak truth table reducibility*; $A \leqslant_{wtt} B$ if there is a computable function k and a Turing reduction of A to B such that for each x, for deciding whether $x \in A$, the queries to B are limited to numbers $\leqslant k(x)$. Sometimes this is called *bounded Turing reducibility.* A consequence of the proof of Theorem 2.1 is that $g \leqslant_{wtt} \varnothing'$ iff f has a computable approximation $g = \lim_s g_s$ where $\#\{s : g_{s+1}(x) \neq g_s(x)\} \leqslant h(x)$ for a computable function h. We would call such an approximation an *ω-computable approximation*, since for each x we compute a finite ordinal $h(x)$ for the mind-change count; then, each time we change our mind about the value of $g(x)$, we need to decrease our pointer. We can extend this to higher ordinals. For example, a $\omega \cdot 2 + 3$-computable approximation $g_s(x)$ would mean that initially we allow 3 mind-changes. If these are exhausted, then we declare a new bound (as large as we like), and so supply ourselves with more mind-changes; and this can happen once more. In terms of ordinals, we start by pointing at the ordinal $\omega \cdot 2 + 2$, then $\omega \cdot 2 + 1$ if we change our mind, then $\omega \cdot 2$ if we change our mind again; if we want to change our mind once more, we point at $\omega + n$ for some n, then $\omega + (n-1)$, $\omega + (n-2)$, ..., until we point at ω; then after another change, we need to point at some finite ordinal k, and from that step onwards, we are allowed no more than k changes, and that number is final. The flexibility in this process is that the numbers n and k are declared *late*; their size can correspond to the stage at which some event happens; and if we do not need all the changes, then k, and perhaps even n, may not need to be declared at all.

The extension to arbitrary ordinals α, at first glance, seems clear. An *α-computable approximation* consists of a computable approximation $\langle g_s \rangle$ of a function g, equipped with a sequence $\langle o_s \rangle$ of counting functions, each $o_s \colon \mathbb{N} \to \alpha$, satisfying $o_{s+1}(x) \leqslant o_s(x)$, and $o_{s+1}(x) < o_s(x)$ if $g_{s+1}(x) \neq g_s(x)$. As well as the sequence $\langle g_s \rangle$, The functions o_s are also required to be uniformly computable. To make sense of this, we cannot work with an abstract "set-theoretic" ordinal α; we need to work with a computable well-ordering of the natural numbers, of order-type α (alternatively we could work with notations for ordinals in the sense of Kleene).

We would then like to define a function to be *α-computably approximable* (or *α-c.a.* for short) if it has an α-computable approximation, for some computable well-ordering of order-type α. This, however, requires some care, as "bad copies" of some ordinals may code too much information, distorting the intended meaning of this notion. Indeed, Ershov showed that under this definition, every Δ^0_2 function is ω-c.a.; the complexity of the function is reflected in a computable copy of ω in which the successor relation is not computable (or if we are working with ordinal notations, a notation for ω^2 in which we cannot computably tell how many copies of ω precede a given point). This points out a limitation of the Spector and Kleene method outside of the Turing degrees (where it is successfully used to define iterations of the Turing jump along computable ordinals).

A perhaps seemingly ad-hoc solution is to restrict ourselved to a class of copies of ordinals in which all copies of a given ordinal are computably isomorphic. This is in general impossible to achieve, but is possible if we restrict ourselves to a proper initial segment of the computable ordinals. For this work, we only deal with ordinals below ϵ_0, where an effective Cantor normal form is sufficient to ensure such "computable categoricity". It may seem that using effective Cantor normal form is an artefact to our studies, but as promised, the resulting hierarchies are robust, characterize the combinatorics of several natural constructions in the literature, and lead to a number of natural definability results. For example, using this approach we will be able to precisely characterize those computably enumerable Turing degrees which bound embeddings of the 5-element modular nondistributive lattice.

Before we move to new material, we will briefly review some of the results mentioned in [16].

3.2. **Degree hierarchies.** Ershov's hierarchy of Δ^0_2 functions is in some sense "orthogonal" to Turing reducibility, in that within the Δ^0_2 degrees, functions which have very low rank in Ershov's hierarchy can have all kinds of Turing degrees, including the greatest Δ^0_2 degree $\mathbf{0}'$. The main idea now is to have a "shotgun wedding" of both notions, as follows:

Definition 3.2. Let $\alpha \leqslant \epsilon_0$.

(1) A c.e. degree $\mathbf{d}$ is *totally α-c.a.* if every $g \in \mathbf{d}$ is α-c.a.
(2) A c.e. degree $\mathbf{d}$ is *totally $< \alpha$-c.a.* if every $g \in \mathbf{d}$ is γ-c.a. for some $\gamma < \alpha$.

We can replace $g \in \mathbf{d}$ by g being any $\mathbf{d}$-computable function, and get the same notion. There is a further notion of *uniformly totally α-c.a.* degrees; this is like the definition of totally α-c.a., except that there is a single $h\colon \mathbb{N} \to \alpha$ such that every function $g \in \mathbf{d}$ has an α-computable approximation $\langle g_s, o_s \rangle$ with $o_s(x) \leqslant h(x)$ for all x. The classes are linearly ordered, as every totally $< \alpha$-c.a. degree is uniformly totally α-c.a.

The case $\alpha = \omega$ is of particular interest, and the definition of totally ω-c.a. degrees was first suggested by Joe Miller. The case of uniformly totally ω-c.a. degrees had been considered earlier, under the name *array computable*[2] (Downey, Jockusch and Stob [26, 27]). The array computable degrees have seen wide applications in computability theory as they characterize the combinatorics of constructions from algorithmic randomness, degree theory and the like.

The following theorem characterises which levels of the hierarchy are proper.

Theorem 3.3 ([22])**.** Let $\alpha < \varepsilon_0$.

(1) There is a totally α-c.a. degree that is not totally γ-c.a. for any $\gamma < \alpha$, if and only if α is closed under addition, i.e. is of the form ω^β for some β. In this case there is, in fact, a uniformly totally α-c.a. degree, which is not totally $< \alpha$-c.a.; and there is a totally α-c.a. degree which is not uniformly so.

(2) There is a totally $< \alpha$-c.a. degree that is not totally γ-c.a. for any $\gamma < \alpha$ iff α is a limit of ordinals that are closed under addition, that is, is of the form ω^β where β is a limit ordinal.

As mentioned above, Nies, Shore and Slaman [55] proved some general metatheorems which allow definability results in the c.e. degrees, via coding models of arithmetic. All their results concern classes that are invariant under taking the double jump, and so cannot be used to define subclasses of the low$_2$ degrees.

Theorem 3.4 ([22])**.** For any α, any totally α-c.a. degree is low$_2$.

Thus, the definability results we soon mention cannot be achieved using these metatheorems. It is tempting to guess that all members of this hierarchy are low. For example if A is superlow (meaning that $A' \equiv_{wtt} \varnothing'$), then A is certainly array computable and hence totally ω-c.a. Moreover, if A' is wtt-reducible to the wtt-jump of $\varnothing'$ then A' is certainly totally ω^2-c.a. However, even the array noncomputable degrees contain non-low c.e. sets (Downey, Jockusch and Stob [26]), and in fact, all levels of the hierarchy contain low sets which are not at lower levels; but no level contains all low c.e. sets. Thus the hierarchy does not align itself with the low sets.

Capturing dynamic combinatorics. As mentioned above, the array computable degrees capture the dynamics of a number of constructions. Two levels of our new hierarchy can be used in this way as well. We give a few examples.

Recall that a real number r is *left-c.e.* if the left cut $\{q \in \mathbb{Q} : q < r\}$ is c.e., iff it is the limit of an increasing computable sequence of rational numbers. One example of such a real is Chaitin's left-c.e. random real, defined as $\Omega = \sum_{U(\sigma)\downarrow} 2^{-|\sigma|}$, where U is a universal prefix-free machine.

In the same way that c.e. sets are halting sets, the theory of algorithmic randomness shows that r is left-c.e. iff it is a *halting probability* (see [24]). Another way of expressing this is that the left-c.e. reals are precisely the measures of the effectively open sets (in either the reals, or Cantor space). A subset of Cantor space is generated by a c.e., equivalently a computable prefix-free set of finite binary strings. Thus, a real r is left-c.e. iff there is some c.e. prefix-free set A of strings such that $r = \sum_{\sigma\in A} 2^{-|\sigma|}$; such a set A is said to *present* r.

[2] Of course, this was not the original definition of array noncomputability, but this version from [27] captures the domination property of the notion in a way that shows the way that it weakens the notion of non-low$_2$-ness, in that $\mathbf{a}$ would be non-low$_2$ using the same Martin's characterization above, but replacing $\leqslant_{wtt}$ by $\leqslant_T$.

Now it is easy to use padding to show that every left-c.e. real has a computable presentation (Downey [19]). On the other hand, bizarre things can happen. In [28], Downey and LaForte showed that there exist a noncomputable left-c.e. real, all of whose c.e. presentations are computable.

Theorem 3.5 ([22])**.** The following are equivalent for a c.e. degree **a**.

(i) **a** is not totally ω-c.a.
(ii) **a** bounds a left-c.e. real r and a c.e. set $B <_T r$ such that if A presents r, then $A \leqslant_T B$.

The proof is quite intricate and uses an elaboration on the "drip feed" strategy of Downey and LaForte [28] for the result mentioned above.

Other theorems relate total ω-c.a.-ness to algorithmic randomness. Recall that a *Martin-Löf* test is a uniformly effectively open (c.e. open) sequence $\{U_n : n \in \omega\}$ with $\mu(U_n) \leqslant 2^{-n}$, where μ denotes the fair-coin measure on Cantor space. A real $A \in 2^\omega$ *passes* the test if $A \notin \cap_n U_n$. If a real passes all Martin-Löf tests then we call it *Martin-Löf random.* Schnorr proved that this is equivalent to $K(A \upharpoonright n) \geqslant^+ n$ for all n, where $\geqslant^+$ means $\geqslant$ up to an additive constant, $A \upharpoonright n$ denotes the first n bits of n, and K denotes prefix-free Kolmogorov complexity.

In the paper [11], Brodhead, Downey and Ng showed that totally ω-c.a. degrees also capture a notion of randomness related to Martin-Löf randomness.

A *Finite Martin-Löf* test is a ML-test with each component U_n clopen; so each U_n is generated by a finite set of strings, but these sets are effectively enumerated, not given in a computable way. We say that the test is *computably bounded* if there is a computable function f such that each U_n is generated by at most $f(n)$ many strings.

In the usual way, these notions gives rise to a notion of randomness, namely finite randomness and computably finite randomness. Note that if we require that the sets of strings are given computably (as finite sets of strings), then we get the notion of Kurtz randomness, see Kurtz [40] or Wang [64]. Also notice that if $A \leqslant_T \varnothing'$ then A is Martin-Löf random iff A is finite random: Say $A = \lim_s A_s$ is not Martin-Löf random; $A \in \cap_n U_n$ for some ML-test $\langle U_n \rangle$. We pause the enumeration of U_n at a stage s at which $A_s \in U_{n,s}$.

Theorem 3.6 (Brodhead, Downey, Ng [11])**.** A c.e. degree **a** contains a computably finite random left-c.e. real iff **a** is not totally ω-c.a.

Proof. (Sketch) The proof is the canonical one. We will enumerate finite sets of open sets $W_{\varphi_e(n)}$ and partial computable functions ψ_e. We ask that $|W_{\varphi_e(n)}| \leqslant \psi_e(n)$. The e^{th} requirement asks that if φ_e, ψ_e are total and the bound is obeyed, then the real A we construct A leaves $[W_{\varphi_e(n)}]$. We will be given some function $g \in \mathbf{a}$ which is not ω-c.a. Each time we see a new σ enumerated into $W_{\varphi_e(n)}[s]$, we will challenge $g(n)[s]$ to change. If it changes, we get permission to leave the test, and can do so rightwards. The fact that **a** is not totally ω-c.a. means that for some argument, g permits at least ψ_e many times. (This only does permitting, coding is easy.) The other direction is symmetric. Each time we need some g we build to change, we cover the real A_s. We refer to [11] for more details. □

There are other relationships with randomness notions: A sequence A is *complex* (see [24])) if $C(A \upharpoonright n) \geqslant h(n)$ for all n, for some order function h. There are several equivalent formalisations, including wtt-computing a diagonally noncomputable function. Every ML-random sequence is complex.

Theorem 3.7 (Ambos-Spies, Fang, Losert, Merkle and Monath [3])**.** A c.e. degree $\mathbf{d}$ is totally ω-c.a. if and only if every left-c.e. real $r \in \mathbf{d}$ is cL-reducible to a complex left-c.e. real, where $A \leqslant_{cL} B$ means that there is a wtt-procedure where the use on argument x is $x + c$ for a fixed c.

Barmpalias, Downey and Greenberg showed that the totally ω-c.a. degrees are related to strong reducibilities and the Cantor-Bendixson rank of reals as follows. The reader may recall that a set A is called *ranked* if A belongs to a countable Π^0_1 class. The following result unifies material from Chisholm et al. [12], Afshari et al. [1], and Barmpalias [8].

Theorem 3.8 (Barmpalias, Downey and Greenberg [9])**.** The following are equivalent for a c.e. degree $\mathbf{a}$:

(1) Every set in $\mathbf{a}$ is wtt-reducible to a ranked set.
(2) Every set in $\mathbf{a}$ is wtt-reducible to a hypersimple set.
(3) Every set in $\mathbf{a}$ is wtt-reducible to a proper initial segment of a computable, scattered linear ordering.
(4) $\mathbf{a}$ is totally ω-c.a.

Moreover, the equivalence still holds if in any of (1), (2) or (3), “set” is replaced by “c.e. set”.

Natural definability. Two further characteriations yield *natural definability* of degree classes in the c.e. degrees, where this means the classes have a definition which, for instance, a lattice-theorist might come up with (contrasted with definitions via coding models of arithmetic). At the present time, as articulated in Shore [59], there are very few such natural definability results.

In [22], and in the prequel [23], we give some new natural definability results for the c.e. degrees. Moreover, these definability results will be related to the central topic of lattice embeddings into the c.e. degrees, analysed by, for instance, Lempp and Lerman [44], Lempp, Lerman and Solomon [45], and Lerman [46].

A central notion for lattice embeddings into the c.e. degrees is the notion of a *weak critical triple.* The reader may recall from Downey [30] and Weinstein [65] that three incomparable elements $\mathbf{a}_0, \mathbf{a}_1$ and $\mathbf{b}$ in an upper semilattice form a weak critical triple if $\mathbf{a}_0 \cup \mathbf{b} = \mathbf{a}_1 \cup \mathbf{b}$ and there is no $\mathbf{c} \leqslant \mathbf{a}_0, \mathbf{a}_1$ with $\mathbf{a}_0 \leqslant \mathbf{b} \cup \mathbf{c}$. We say that incomparable $\mathbf{a}_0, \mathbf{a}_1$ and $\mathbf{b}$ in an upper semilattice form a *critical triple* if $\mathbf{a}_0 \cup \mathbf{b} = \mathbf{a}_1 \cup \mathbf{b}$ and every $\mathbf{c}$ below both $\mathbf{a}_0$ and $\mathbf{a}_1$ is also below $\mathbf{b}$.[3] These notions capture the need for “continuous tracing” which is used in an embedding of the lattice M_5 (Fig. 1) into the c.e. degrees (first embedded by Lachlan [41]).

Indeed the first nonembeddability result was by Lachlan and Soare [43] who demonstrated that an “infimum into an M_5” could not be embedded in the c.e. degrees by showing that the lattice S_8 (Fig. 2) could not be embedded (as suggested by Lerman).

Downey, Greenberg and Weber proved the following:

Theorem 3.9 ([23])**.** The following are equivalent for a c.e. degree $\mathbf{a}$:

[3] This definition becomes more natural in a lattice, where we can write $\mathbf{a}_0 \cap \mathbf{a}_1 \leqslant \mathbf{b}$. We recall that a finite lattice is join semidistributive iff it is principally indecomposable iff it contains no copy of M_5 iff it contains no critical triple iff it contains no weak critical triple. See [45].

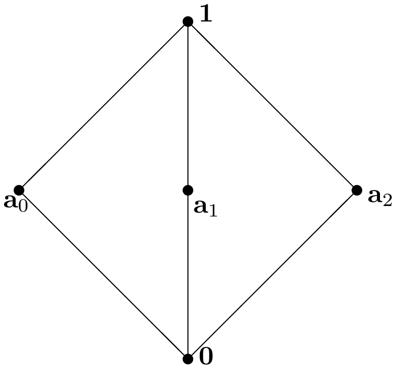

FIGURE 1. The 1-3-1 lattice

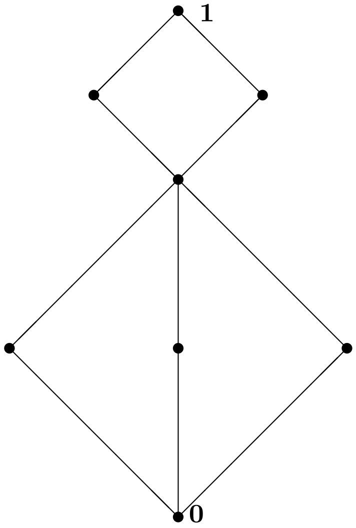

FIGURE 2. The lattice S_8

(1) $\mathbf{a}$ bounds a critical triple in the c.e. degrees;
(2) $\mathbf{a}$ bounds a weak critical triple in the c.e. degrees;
(3) $\mathbf{a}$ is not totally ω-c.a.

Hence the class of totally ω-c.a. degrees is naturally definable in the c.e. degrees.

For embeddings of M_5 we need a higher level of permitting:

Theorem 3.10 ([22])**.** A c.e. degree $\mathbf{a}$ is not totally $< \omega^\omega$-c.a. iff $\mathbf{a}$ bounds a copy of M_5 in the c.e. degrees.

These proofs use constructions which involve high levels of nonuniformity. They are discussed in detail in [16]. Ambos-Spies and Losert [5] have given an example of a 7 element lattice (Fig. 3) which is embeddable in the c.e. degrees precisely below degrees which are not totally ω-c.a.

The following question remains open:

Question 3.11. Is there an $n > 1$ such that the class of totally ω^n-c.a. degrees is (naturally) definable in the c.e. degrees?

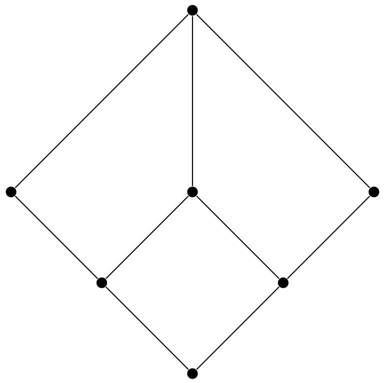

FIGURE 3. The lattice L_7

Li Ling Ko [38] has a number of results related to this question. She examined the question for a number of lattices and showed that all fall into either the M_5 case or into the L_7, case. Cholak and her conjecture that all lattices embeddable into the c.e. degrees fall into these two classes, so that lattices cannot be used to answer Question 3.11 above.

3.3. **More generally.** There have been some attempts to understand the hierarchy outside of the c.e. degrees. As with array computability, the natural notion to consider here is *domination* rather than approximation. For eample, a Turing degree $\mathbf{d}$ is *α-c.a. dominated* if every $g \in \mathbf{d}$ is dominated by some α-c.a. function; a c.e. degree is α-c.a. dominated if and only if it is totally α-c.a. There are similar domination properties corresponding to uniform totally α-c.a. degrees and totally $< \alpha$-c.a. degrees.

Michael McInerney [48] has demonstrated other connections between genericity and our hierarchy. He studied notions of *multiple genericity* related to pb-genericity of Downey, Jockusch and Stob [27]. In particular, he defines a notion of ω-change genericity, a strengthening of pb-genericity. A Turing degree bounds a pb-generic sequence if and only if it is array noncomputable.

Theorem 3.12 (McInerney [48])**.** A c.e. degree bounds an ω-change generic sequence if and only if it is not totally ω-c.a.

Note though that the characterisation mentioned above of bounding pb-generics holds for all Turing degrees, not only the c.e. ones.

Theorem 3.13 (McInerney [48])**.** Let $\mathbf{d}$ be a Turing degree.

(1) If $\mathbf{d}$ is not uniformly ω^2-c.a. dominated then $\mathbf{d}$ computes an ω-change generic sequence.
(2) If $\mathbf{d}$ is ω-c.a. dominated then it does not compute an ω-change generic sequence.

Whether (1) can be improved to ω-c.a. domination remains open.

Recently, McInerney and Ng extended these studies and introduced a transfinite hierarchy of genericity notions stronger than 1-genericity and weaker than 2-genericity. There are many connections with the hierarchy of totally ω-c.a. degrees. In [50], McInerney and Ng give several theorems concerning the strength required to compute multiply generic degrees, and show that some of the levels in the hierarchy can be separated, and that these separations can be witnessed by a Δ^0_2 degree. In that paper, they consider downward density for these classes.

In a later paper, McInerney and Ng investigated the relationship between analogs of weak genericity and genericity.

Theorem 3.14 (McInerney and Ng [49])**.** Let $\alpha < \varepsilon_0$. There is a Δ^0_2 degree $\mathbf{a}$ which is weakly α-change generic but not α-change generic.

It would be interesting to understand the analog of the hierarchy outside of the Δ^0_2 degrees via domination properties.

4. Sacks Splitting Theorem

Our methods can be used to understand some classical constructions in computability theory. One such construction is Sacks's Splitting Theorem. This theorem asserts that if A is c.e., then there exist disjoint c.e. sets $A_1 \sqcup A_2 = A$ with $A_1|_T A_2$. Ever since Soare's classic paper [60], the Sacks Splitting Theorem is pointed to as a quintessential example of a finite injury argument of "unbounded type". By this we mean the following. The standard simple proof of the existence of a c.e. set of low degree, of the Friedberg-Muchnik Theorem, as per Soare's book [61] (or any other standard text), uses a finite injury priority argument where requirements are injured at most a computable number of times. In the standard proof of the Friedberg-Muchnik Theorem each requirement R_{2e} is injured at most 2^e many times. This makes the relevant sets not only low, but superlow ($A' \equiv_{wtt} \varnothing'$), since each A-computable partial function can be computed with an approximation with at most a computable number of mind-changes in the sense of the limit lemma.

We would argue that we can measure how "unbounded" the construction is using the Downey-Greenberg hierarchy. We have the following results:

Theorem 4.1 (Downey and Ng [29])**.** There is a c.e. degree $\mathbf{a}$ such that if $\mathbf{a_0} \vee \mathbf{a_1} = \mathbf{a}$ in the c.e. degrees, then one of $\mathbf{a_0}$ or $\mathbf{a_1}$ is not totally ω-c.a.

Downey and Ng also showed that every high c.e. degree is the join of two totally ω-c.a. c.e. degrees. This extends a classical theorem of Bickford and Mills [10] who showed that $\mathbf{0}'$ is the join of two superlow c.e. degrees. In [29] it is *also* shown that there are (super-)high c.e. degrees that are not the joins of two superlow degrees.

Thus if we use the Downey-Greenberg hierarchy for the classification of the complexity of the splits, we cannot hope to do better than ω^2. The classical Sacks' construction shows that it gives that a c.e. set A can be split into a pair of low, totally ω^ω-c.a. c.e. sets. This can be improved:

Theorem 4.2 (Ambos-Spies, Downey, Monath and Ng [2])**.** Every c.e. set can be split into a pair of low c.e. sets which are totally ω^2-c.a.

The proof of this recent result is too complex to include here, but to give the reader some idea of the dynamics of the methodology, we will give a proof of a weaker result, obtained earlier by Ambos-Spies, Downey and Monath. The ideas are similar but Theorem 4.2 heavily also uses some priority tree machinery, hence obscuring some of the main ideas.

Theorem 4.3. Every c.e. set can be split into a pair of low c.e. sets which are totally ω^3-c.a.

Proof. We represent the ordinal $c_2 \cdot \omega^2 + c_1 \cdot \omega + c_0 < \omega^3$ by the triple (c_2, c_1, c_0). So the ordering on the ordinals $< \omega^3$ coincides with the lexicographical ordering on $\omega \times \omega \times \omega$.

W.l.o.g. assume that A is infinite, fix a 1-1 computable function a enumerating A, and let $A_s = \{a(t) : t < s\}$. The sets A_0 and A_1 are enumerated in stages where, at stage $s+1$, $a(s)$ is put into either A_0 or A_1. So $A_{i,s} = \{a(t) : t < s \;\&\; a(t) \in A_i\}$ is the finite part of A_i enumerated by the end of stage s; so in the end, $A = A_0 \sqcup A_1$, and by effectivity of the construction, both A_0 and A_1 are c.e. In order to ensure that the sets A_0 and A_1 are totally ω^3-c.a., it suffices to meet the global requirements

$$\mathcal{R}^{global}_{i,e} : \text{ If } \Phi^{A_i}_e \text{ is total then } \Phi^{A_i}_e \text{ is } \omega^3\text{-c.a.}$$

for $i \leqslant 1$ and $e \geqslant 0$. For this sake we have a list $\{\mathcal{R}_n\}_{n \geqslant 0}$ of requirements, where the primary goal of requirement $\mathcal{R}_{2\langle e,x\rangle+i}$ is to preserve convergent computations $\Phi^{A_{i,s}}_{e,s}(x)$ by restraining numbers $< \varphi^{A_{i,s}}_e(x)$ from A_i. We say that $\mathcal{R}_{2\langle e,x\rangle+i}$ is an *i-requirement* and an *i-e-requirement.* We let $r(n,s)$ (which will be specified below) be the restraint imposed by $\mathcal{R}_n$ at stage $s+1$ and we let

$$R(n,s) = \max_{n' \leqslant n, t \leqslant s} r(n',t)$$

be the corresponding accumulated restraint. (Working with R in place of r is for technical convenience.) Then, at stage $s+1$, $a(s)$ is put into A_1 if the least n such that $a(s) < R(n,s)$ is a 0-requirement (or if no such n exists), and $a(s)$ is put into A_0 otherwise. Moreover, we say that an i-requirement $\mathcal{R}_n$ is *injured* at stage $s+1$ if $a(s) < R(n,s)$ and $a(s)$ is enumerated into A_i. (Note that if the i-requirement $\mathcal{R}_n$ is injured at stage $s+1$ then all lower priority i-requirements $\mathcal{R}_{2m+i}$ $(n < 2m+i)$ are injured too and there is a higher priority $(1-i)$-requirement $\mathcal{R}_{2m'+(1-i)}$ $(2m' + (1-i) < n)$ such that $a(s) < R(2m' + (1-i), s)$.)

Now, if we would let $r(2\langle e,x\rangle + i, s) = r_p(2\langle e,x\rangle + i, s)$ for

$$r_p(2\langle e,x\rangle + i, s) = \begin{cases} \varphi^{A_{i,s}}_e(x) & \text{if } \Phi^{A_{i,s}}_{e,s}(x)\downarrow \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

then (modulo the minor technical change that here we work with the accumulated restraint R in place of the proper restraint r) we would obtain a construction yielding totally ω^ω-c.a. sets A_0 and A_1. In the following we call $r_p(n,s)$ defined according to (1) the *primary restraint* of $\mathcal{R}_n$ at stage s, and we tacitly assume that the restraint function r is chosen so that $r(n,s) \geqslant r_p(n,s)$ for all $n, s \geqslant 0$.

Now, the key idea which will enable us to reduce the complexity of the sets A_0 and A_1 in the Downey-Greenberg hierarchy is as follows. In certain situations we will *link* an i-requirement $\mathcal{R}_{n+2}$ to a higher priority i-requirement $\mathcal{R}_{n'}$ $(n' \leqslant n)$. This will mean that $\mathcal{R}_{n'}$ (in addition to its primary task) will take over the task of $\mathcal{R}_{n+2}$ by imposing (in addition to its own primary restraint $r_p(n', s)$) the primary restraint $r_p(n+2, s)$ of $\mathcal{R}_{n+2}$ on A_i. The technical observation which will guide us when we will build a link is the following.

Assume that we define the restraints by letting $r(n,s) = r_p(n,s)$. Then there are uniformly computable functions $m_{[n,s]} : \omega \to \omega^3$ $(n, s \geqslant 0)$ such that, for any numbers n, s_0 and s_1, $m_{[n,s_0]}(s_0) = m_{[n,s_1]}(s_1)$ and the following holds. If $s_0 < s_1$ and, for any stage s with $s_0 \leqslant s < s_1$, no requirement $\mathcal{R}_{n'}$ with $n' \leqslant n$ is injured at stage $s+1$, then $m_{[n,s_0]}(s)$ is nonincreasing on $[s_0, s_1)$, i.e., $m_{[n,s_0]}(s+1) \leqslant m_{[n,s_0]}(s)$

for s with $s_0 \leqslant s < s_1$, and, for any such s such that $\mathcal{R}_{n+2}$ becomes injured at stage $s+1$, $m_{[n,s_0]}(s+1) < m_{[n,s_0]}(s)$. (This observation is nontrivial and will be implicitly proven in the stronger Claim 5 below.) In fact, it is crucial that, for obtaining such functions, it suffices that the restraint function r satisfies the *restraint condition* which requires that there is a computable bound $f(n)$ on the number $R(n,s)$ may change (i.e., grow) on any interval of stages at which neither $\mathcal{R}_n$ nor a higher priority requirement is injured, i.e., for any index n and any stages $s < s'$ such that there is no stage t and no index $n' \leqslant n$ such that $s \leqslant t < s'$ and requirement $\mathcal{R}_{n'}$ is injured at stage $t+1$,

$$f(n) \geqslant |\{t : s \leqslant t < s' \;\&\; R(n,t+1) > R(n,t)\}|. \tag{2}$$

The above leads to the following (rough) idea. Fix e and i such that $\Phi_e^{A_i}$ is total and consider the canonical approximation ψ of $\Phi_e^{A_i}$. Then any change $\psi(x,s+1) \neq \psi(x,s)$ implies that the (primary restraint of the) requirement $\mathcal{R}_{2\langle e,x\rangle+i}$ is injured at stage $s+1$. So, if we let $n+2 = 2\langle e,x\rangle + i$ (w.l.o.g. we may may assume that $2\langle e,x\rangle + i > 2$) then, on any interval $[s_0, s_1)$ of stages such that no requirement $\mathcal{R}_{n'}$ with $n' \leqslant n$ is injured at any stage $t+1$ with $s_0 \leqslant t < s_1$, the function $m_{[n,s_0]}$ *locally* witnesses that $\Phi_e^{A_i}$ is ω^3-c.a. via ψ. Now, the idea is to get a global witness by piecing together such local witnesses as follows. If an i-requirement $\mathcal{R}_{n'}$ with $n' \leqslant n$ or a higher priority requirement $\mathcal{R}_{n''}$ ($n'' < n'$) becomes injured at stage $s+1$ then requirement $\mathcal{R}_{n+2}$ becomes *linked* to $\mathcal{R}_{n'}$. This will give the desired ω^3-approximation as follows. Fix the numbers $n_0 > n_1 > \cdots > n_k$ such that, for some stage s, n_j is minimal such that $\mathcal{R}_{n+2}$ is linked to $\mathcal{R}_{n_j}$ at stage s, and let s_j be the stage at which $\mathcal{R}_{n+2}$ becomes linked to $\mathcal{R}_{n_j}$ for the first time (where we assume that $\mathcal{R}_{n+2}$ is linked to itself at stage 0, i.e., $n_0 = n+2$ and $s_0 = 0$; moreover, w.l.o.g. we may assume that the first two requirements are never injured whence $n_k \geqslant 2$). Then $s_0 > s_1 > \cdots > s_k$ and, for $s \in [s_j, s_{j+1})$ (where $j \leqslant k$ and $s_{k+1} = \omega$), $m_{[n_j-2,s_j]}$ locally witnesses that $\Phi_e^{A_i}$ is ω^3-c.a. via ψ. So the computable function $m : \omega \times \omega \to \omega^3$ where, for any $j \leqslant k$ and any $s \in [s_j, s_{j+1})$,

$$m(x,s) = \left(\sum_{n' < n_j - 2} (m_{[n',0]}(0) + 1) \right) + m_{[n_j-2,s_j]}(s)$$

globally witnesses that $\Phi_e^{A_i}$ is ω^3-c.a. via ψ (for moving from $m_{[n_j-2,s_j]}$ to $m_{[n_{j+1}-2,s_{j+1}]}$ at stage s_{j+1}, note that $m_{[n_{j+1}-2,s_{j+1}]}(s_{j+1}) = m_{[n_{j+1}-2,0]}(0)$ hence $m(x,s_{j+1}) < m(x,s_{j+1}-1)$; see the proof of Claim 6 below for details).

If we link requirements and define the restraints $r(n,s)$ as described above, however, the restraint function will not satisfy the restraint condition whence we cannot argue that the required functions $m_{[n,s]}$ will exist. To overcome this problem we will take some precautions when building links.

First, at any stage $s+1$ at which requirement $\mathcal{R}_n$ is injured, only finitely many requirements may become linked to $\mathcal{R}_n$ (e.g. only requirements with index $\leqslant s$). This will allow us to argue that only finitely many requirements will ever be linked to $\mathcal{R}_n$, that the restraint of $\mathcal{R}_n$ will be bounded, and that $\mathcal{R}_n$ will be injured only finitely often (see Claim 2 below). Next, we link a requirement $\mathcal{R}_{n'}$ ($n' > n$) to the i-requirement $\mathcal{R}_n$ only if $\mathcal{R}_{n'}$ is an i-e-requirement where $e < n$. (Note that this restriction is not serious since, for fixed i and e, it may happen only finitely often that a requirement $\mathcal{R}_n$ with $n \leqslant e$ is injured. So we may use the least stage after which this does not happen anymore in the definition of the ω^3-approximation of $\Phi_e^{A_i}$.) This ensures that the requirements linked to $\mathcal{R}_n$ work on $\leqslant n$ global requirements $\mathcal{R}_{i,e}^{global}$. Finally, since a global requirement $\mathcal{R}_{i,e}^{global}$ is trivially met if the corresponding function $\Phi_e^{A_i}$ is not total, we can combine the restraints of the local requirements $\mathcal{R}_{2\langle e,x\rangle+i}$ linked to $\mathcal{R}_n$ which work on $\mathcal{R}_{i,e}^{global}$ in *one* restraint by *delaying* the definition of this restraint to the first stage s at which $\Phi_{e,s}^{A_{i,s}}(x)$ is defined for all x such that the requirement $\mathcal{R}_{2\langle e,x\rangle+i}$ was linked to $\mathcal{R}_n$ at the greatest stage $< s$ at which $\mathcal{R}_n$ was injured. (This delay in imposing the restraint requires that the approximation ψ of $\Phi_e^{A_i}$ has to be adjusted correspondingly.) Then, besides the proper restraint of $\mathcal{R}_n$, there are at most n additional restraints for the sake of the requirements linked to $\mathcal{R}_n$, and each of these restraints may grow at most once unless $\mathcal{R}_n$ becomes injured. Obviously, this guarantees the restraint condition.

Having explained the key ideas, we now formally describe the construction. Links and restraints at stage s are defined by a simultaneous induction on s. We first give the linking procedure and then specify the restraint function.

Given $m \geqslant 0$ and $i \leqslant 1$ fix $e \geqslant 0$ such that $\mathcal{R}_{2m+i}$ is an i-e-requirement. $\mathcal{R}_{2m+i}$ becomes *linked* to itself at stage 0 (we call this an *unproper link*). Moreover, if there is a stage $s \geqslant 2m+i$ and an i-requirement $\mathcal{R}_{2m'+i}$ such that $e < m' < m$ and $\mathcal{R}_{2m'+i}$ or a higher priority requirement is injured at stage $s+1$ then $\mathcal{R}_{2m+i}$ becomes *(properly) linked* to $\mathcal{R}_{2m'+i}$ at stage $s+1$. Finally, once $\mathcal{R}_{2m+i}$ becomes linked to a requirement $\mathcal{R}_{2m'+i}$ it remains linked to it forever.

Note that a requirement may be linked to more than one requirement. Let $l(n,s)$ be the index of the highest priority requirement to which $\mathcal{R}_n$ becomes linked by the end of stage s,

$$l(n,s) = \mu\, n' \, [\mathcal{R}_n \text{ is linked to } \mathcal{R}_{n'} \text{ at stage } s].$$

Note that $l(n,0) = n$, $l(n,s+1) \leqslant l(n,s) \leqslant n$ and $2 \leqslant l(n,s)$ unless $l(n,s) = n$. Conversely, for an i-requirement $\mathcal{R}_{2m+i}$ and $e' < m$, let

$$\mathrm{Link}(2m+i, e', s) = \{x' : \ \mathcal{R}_{2\langle e',x'\rangle+i} \text{ is linked to } \mathcal{R}_{2m+i} \text{ at stage } s\}.$$

Note that

$$\bigcup_{e'<m} \{2\langle e',x'\rangle + i : x' \in \mathrm{Link}(2m+i,e',s)\} \subseteq \omega \restriction s$$

is the set of the indices of the requirements that are *properly* linked to $\mathcal{R}_{2m+i}$ at the end of stage s.

Let $\mathcal{R}_n = \mathcal{R}_{2m+i} = \mathcal{R}_{2\langle e,x\rangle+i}$ be an i-e-requirement. The *restraint* $r(n,s)$ imposed by $\mathcal{R}_n$ on A_i at stage $s+1$ is inductively defined as follows. $r(n,0) = 0$. For $s > 0$,

$$r(n,s) = \max_{e' \leqslant m} r_{e'}(n,s)$$

where the e'th subrestraint $r_{e'}(n,s)$ is defined as follows ($e' \leqslant m$). If $e' = m$ then $r_{e'}(n,s) = r_p(n,s)$ where $r_p(n,s)$ is the primary restraint of $\mathcal{R}_n$ defined according to (1). If $e' < m$ distinguish the following two cases. If $\mathcal{R}_n$ is injured at stage s then $r_{e'}(n,s) = 0$. Otherwise, for $e' < m$, $r_{e'}(n,s)$ is the maximum restraint which the i-e'-requirements linked to $\mathcal{R}_n$ at stage s would like to impose in order to protect their corresponding computations - provided that *all* of these computations are defined. To be more precise,

$$r_{e'}(n,s) = \max_{y \in \mathrm{Link}(n,e',s)} \varphi_{e'}^{A_{i,s}}(y)$$

if Link(n,e',s) is nonempty and $\Phi_{e',s}^{A_{i,s}}(y)\downarrow$ for all $y \in$ Link(n,e',s), and

$$r_{e'}(n,s) = 0$$

otherwise.

(Note that at stage 0 just the unproper links exist. The links defined at stage $s+1$ depend only on the restraints $r(n,s)$ defined by the end of stage s (namely, whether a requirement $\mathcal{R}_n$ is injured at stage $s+1$ or not depends only on the restraint $R(n,s)$ defined at stage s). Finally, the restraints $r(n,s)$ defined at stage s depend only on the links defined by this stage. So the above definitions are sound.)

This completes the construction. In the remainder of the proof we show that the sets A_0 and A_1 have the desired properties. Obviously, $A = A_0 \cup A_1$, A_0 and A_1 are disjoint, and, by effectivity of the construction, A_0 and A_1 are c.e. So it suffices to show that the global requirements are met and that the sets A_0 and A_1 are low. For this sake we prove a series of claims.

Claim 1. Let $n = 2m + i \geqslant 0$ (where $i \leqslant 1$) and let s_0 and s_1 be stages such that $s_0 < s_1$ and such that there is no stage s such that $s_0 \leqslant s < s_1$ and $\mathcal{R}_n$ is injured at stage $s+1$. Then there is no requirement newly linked to $\mathcal{R}_n$ at any stage $s+1$ such that $s_0 \leqslant s < s_1$, hence

$$\forall\, e' < m\ \forall\, s\ (s_0 \leqslant s < s_1 \ \Rightarrow\ \mathrm{Link}(n,e',s+1) = \mathrm{Link}(n,e',s_0)). \tag{3}$$

Moreover, the restraint $r(n,s)$ changes at most $n+1$ times after stage s_0 and before s_1, i.e.,

$$|\{s : s_0 \leqslant s < s_1 \ \&\ r(n,s+1) \neq r(n,s)\}| \leqslant n+1. \tag{4}$$

Proof. By assumption,

$$\forall\, s\ (s_0 \leqslant s < s_1 \ \Rightarrow\ (A_{i,s+1} \upharpoonright R(n,s) = A_{i,s} \upharpoonright R(n,s)). \tag{5}$$

Since a new requirement becomes linked to $\mathcal{R}_n$ only at a stage where $\mathcal{R}_n$ is injured, (3) is immediate. For a proof of (4) note that $r(n,s+1) \neq r(n,s)$ implies that $r_{e'}(n,s+1) \neq r_{e'}(n,s)$ for at least one of the $m+1$ numbers $e' \leqslant m$. So, by $m \leqslant n$, given $e' \leqslant m$ and a stage s such that $s_0 \leqslant s < s_1$ and $r_{e'}(n,s) > 0$, it suffices to show that $r_{e'}(n,s+1) = r_{e'}(n,s)$. This is done as follows. If $e' < m$ then, by $r_{e'}(n,s) > 0$, for any $y \in$ Link(n,e',s), the corresponding computation $\Phi_{e',s}^{A_{i,s}}(y)$ converges and its use is bounded by $r_{e'}(n,s)$. Since $r_{e'}(n,s) \leqslant R(n,s)$, it follows by (5) that none of these computations is injured at stage $s+1$. So $r_{e'}(n,s+1) = r_{e'}(n,s)$ by (3). Finally, if $e' = m$ then, for the unique e and x such that $m = \langle e,x\rangle$, $\Phi_{e,s}^{A_{i,s}}(x)\downarrow$ and $\varphi_e^{A_{i,s}}(x) = r_m(n,s) \leqslant R(n,s)$. So, by (5), the computation is preserved at stage $s+1$ hence $r_m(n,s+1) = r_m(n,s)$.

This completes the proof of Claim 1.

Claim 2. Let $n = 2m + i \geqslant 0$ (where $i \leqslant 1$). Requirement $\mathcal{R}_n$ is injured at most finitely often, for any $e' < m$,

$$\lim_{s \to \infty} \mathrm{Link}(n, e', s)$$

exists and is finite, and

$$\lim_{s \to \infty} r(n, s) < \omega$$

exists.

Proof. The proof is by induction. Fix n and, by inductive hypothesis, assume the claim for $n' < n$. Then there is a stage s_0 and a number u, such that $A \upharpoonright u = A_{s_0} \upharpoonright u$, and, for all $s \geqslant s_0$ and $n' < n$, $\mathcal{R}_{n'}$ is not injured at stage s and $R(n', s) = R(n', s_0) \leqslant u$. So, for any $s \geqslant s_0$ and any n'' such that $a(s) < R(n'', s)$, it holds that $n'' \geqslant n$. It follows by construction that $\mathcal{R}_n$ is not injured after stage s_0. The other parts of the claim follow by Claim 1.

The next claim shows that the restraint condition is satisfied via the computable bound $f(n) = (n + 1)^2$.

Claim 3. Let $n \geqslant 0$ and let s_0 and s_1 be stages such that $s_0 < s_1$ and such that, for any stage s with $s_0 \leqslant s < s_1$, no requirement $\mathcal{R}_{n'}$ with $n' \leqslant n$ is injured at stage $s + 1$. Then

$$|\{s : s_0 \leqslant s < s_1 \;\&\; R(n, s + 1) \neq R(n, s)\}| \leqslant (n + 1)^2. \tag{6}$$

Proof. Note that $R(n, s + 1) \neq R(n, s)$ implies that $r(n', s + 1) \neq r(n', s)$ for some $n' \leqslant n$. So Claim 3 is immediate by Claim 1.

Claim 4. Let $n \geqslant 0$ and let s_0 and s_1 be stages such that $s_0 < s_1$ and such that $R(n, s) = R(n, s_0)$ for all stages s with $s_0 \leqslant s \leqslant s_1$. Then there are at most $R(n, s_0)$ stages s with $s_0 \leqslant s < s_1$ such that $\mathcal{R}_{n+1}$ is injured at stage $s + 1$ and

$$|\{s : s_0 \leqslant s < s_1 \;\&\; R(n + 1, s + 1) \neq R(n + 1, s)\}| \leqslant (n + 2) \cdot R(n, s_0). \tag{7}$$

Proof. Note that if $\mathcal{R}_{n+1}$ is injured at a stage $s + 1 > s_0$ such that $R(n, s) = R(n, s_0)$ then $a(s) < R(n, s_0)$. So this can happen at most $R(n, s_0)$ times. Correctness of (7) follows by Claim 1 since, for s as in (7), $R(n, s + 1) = R(n, s)$ hence $r(n + 1, s + 1) \neq r(n + 1, s)$.

Claim 5. Let $n \geqslant 0$, let s_0 and s_1 be stages such that $s_0 < s_1$ and such that, for any stage s with $s_0 \leqslant s < s_1$, no requirement $\mathcal{R}_{n'}$ with $n' \leqslant n$ is injured at stage $s + 1$, let

$$(m_2(s_0), m_1(s_0), m_0(s_0)) = ((n + 1)^2 + 1, 0, 0), \tag{8}$$

and, for s such that $s_0 \leqslant s$, let

$$m_2(s + 1) = (n + 1)^2 - |\{s' : s_0 \leqslant s' \leqslant s \;\&\; R(n, s' + 1) \neq R(n, s')\}|, \tag{9}$$

(10)

$$\begin{aligned} m_1(s + 1) \quad = \quad & (n + 2) \cdot R(n, (s + 1)^-) \\ & - |\{s' : (s + 1)^- \leqslant s' \leqslant s \;\&\; R(n + 1, s' + 1) \neq R(n + 1, s')\}| \end{aligned}$$

where $(s+1)^- = \mu\, t \geqslant s_0\, [R(n,t) = R(n,s+1)]$*, and*

$$m_0(s+1) = R(n+1,s+1) - |A_{s+1} \restriction R(n+1,s+1)| \tag{11}$$

Then the following hold.

(i) For $j \leqslant 2$ *and* $s_0 \leqslant s \leqslant s_1$*,* $m_j(s) \geqslant 0$*.*

(ii) For $s_0 \leqslant s < s_1$*,*

$$(m_2(s+1), m_1(s+1), m_0(s+1)) \leqslant (m_2(s), m_1(s), m_0(s))$$

(with respect to the lexicographical ordering on $\omega \times \omega \times \omega$*).*

(iii) If $s_0 \leqslant s < s_1$ *and requirement* $\mathcal{R}_{n+2}$ *is injured at stage* $s+1$ *then*

$$(m_2(s+1), m_1(s+1), m_0(s+1)) < (m_2(s), m_1(s), m_0(s)). \tag{12}$$

Proof. (i). For $s = s_0$ the claim is immediate by (8). So, given $j \leqslant 2$ and s with $s_0 \leqslant s < s_1$, it suffices to show that $m_j(s+1) \geqslant 0$. For $j = 0$, this is obvious. For $j = 1$ and $j = 2$ this follows by Claim 4 and Claim 3, respectively.

(ii). Fix s such that $s_0 \leqslant s < s+1 \leqslant s_1$. It suffices to show the following.

(a) $m_2(s+1) \leqslant m_2(s)$
(b) $m_2(s+1) = m_2(s) \;\Rightarrow\; m_1(s+1) \leqslant m_1(s)$
(c) $m_2(s+1) = m_2(s)\ \&\ m_1(s+1) = m_1(s) \;\Rightarrow\; m_0(s+1) \leqslant m_0(s)$

Claim (a) is obvious. For a proof of (b) assume that $m_2(s+1) = m_2(s)$. Then $s_0 < s$ and $R(n,s+1) = R(n,s)$. By the former, $m_1(s)$ and $m_1(s+1)$ are defined according to (10) while, by the latter, $(s+1)^- = s^-$. Obviously, this implies $m_1(s+1) \leqslant m_1(s)$. Finally, for a proof of (c) assume that $m_2(s+1) = m_2(s)$ and $m_1(s+1) = m_1(s)$. Then, by the former, $s_0 < s$, $R(n,s+1) = R(n,s)$ and $(s+1)^- = s^-$. By $m_1(s+1) = m_1(s)$ and by (10), this implies that $R(n+1,s+1) = R(n+1,s)$. Since $m_0(s)$ and $m_0(s+1)$ are defined according to (11) and since A_s is contained in A_{s+1}, this shows that $m_0(s+1) \leqslant m_0(s)$.

(iii). Assume that $s_0 \leqslant s < s_1$ and requirement $\mathcal{R}_{n+2}$ is injured at stage $s+1$. Then $a(s) < R(n+1,s)$. Moreover, by (ii) it suffices to show that $m_j(s+1) \neq m_j(s)$ for some $j \leqslant 2$. If $s = s_0$ then, obviously, $m_2(s+1) < m_2(s)$. So w.l.o.g. we may assume that $s > s_0$ whence $m_2(s)$, $m_1(s)$ and $m_0(s)$ are defined according to (9), (10) and (11), respectively. The claim follows, by distinguishing the following three cases. If $R(n,s+1) \neq R(n,s)$ then $m_2(s+1) < m_2(s)$ by (9); if $R(n,s+1) = R(n,s)$ (hence $(s+1)^- = s^-$) and $R(n+1,s+1) \neq R(n+1,s)$ then $m_1(s+1) < m_1(s)$ by (10); and if $R(n,s+1) = R(n,s)$ and $R(n+1,s+1) = R(n+1,s)$ then $m_0(s+1) < m_0(s)$ by (11) and by $a(s) < R(n+1,s)$.

This completes the proof of Claim 5.

The next claim shows that A_0 and A_1 are totally ω^3-c.a.

Claim 6. For any $e \geqslant 0$ *and* $i \leqslant 1$*, the global requirement* $\mathcal{R}^{global}_{i,e}$ *is met.*

Proof. Fix e and i such that w.l.o.g. $\Phi_e^{A_i}$ is total. It suffices to define total computable functions $\psi : \omega \times \omega \to \omega$ and $m_j : \omega \times \omega \to \omega$ $(j \leqslant 2)$ such that, for all $x \geqslant 0$,

$$\lim_{s\to\infty} \psi(x,s) = \Phi_e^{A_i}(x) \tag{13}$$

and, for all $x, s \geqslant 0$,

$$(m_2(x,s+1), m_1(x,s+1), m_0(x,s+1)) \leqslant (m_2(x,s), m_1(x,s), m_0(x,s)) \tag{14}$$

and

$$\begin{gathered} \psi(x,s+1) \neq \psi(x,s) \\ \Rightarrow \\ (m_2(x,s+1), m_1(x,s+1), m_0(x,s+1)) < (m_2(x,s), m_1(x,s), m_0(x,s)) \end{gathered} \tag{15}$$

hold.

By Claim 2 fix s^* minimal such that, for all $n' < \max\{2, e+1\}$, $A \upharpoonright R(n', s^*) = A_{s^*} \upharpoonright R(n', s^*)$ and, for $s \geqslant s^*$, $\mathcal{R}_{n'}$ is not injured at stage s and $r(n',s) = r(n',s^*)$ (hence $R(n',s) = R(n',s^*)$). Moreover, for $x \leqslant 1$ and $s \geqslant 0$, let $\psi(x,s) = \Phi_e^{A_i}(x)$ and $m_2(x,s) = m_1(x,s) = m_0(x,s) = 0$. For $x \geqslant 2$ and $s \geqslant 0$, let $n_s^x = l(2\langle e,x\rangle + i, s)$ be the index of the highest priority (i-)requirement to which $\mathcal{R}_{2\langle e,x\rangle+i}$ is linked to at stage s. Note that (by $x \geqslant 2$ and by construction)

$$2\langle e,x\rangle + i = n_0^x \geqslant n_s^x \geqslant n_{s+1}^x \geqslant 2$$

for all $s \geqslant 0$. So, in particular, there is a number n^x such that $n_s^x = n^x$ for all sufficiently large stages s. It follows, by totality of $\Phi_e^{A_i}$ and by (the second part of) Claim 2, that, for almost all stages s,

$$\Phi_{e,s}^{A_{i,s}}(x)\downarrow \ \& \ r(n_s^x, s) \geqslant \varphi_e^{A_{i,s}}(x) \tag{16}$$

holds. Now, for $x \geqslant 2$ and $s \geqslant 0$, let

$$\psi(x,s) = \Phi_{e,s'}^{A_{i,s'}}(x) \quad \begin{array}{l} \text{for } s' \text{ minimal such that } s' \geqslant \max\{s, 2\langle e,x\rangle + i + 1, s^*\} \\ \text{and (16) holds (for } s' \text{ in place of } s). \end{array}$$

Obviously, ψ is total and computable and (13) holds.

Fo the definition of $m_j(x,s)$ $(j \leqslant 2)$ for $x \geqslant 2$ distinguish the following two cases. For $s \geqslant s^*$ fix $s_0 \leqslant s$ minimal such that $s^* \leqslant s_0$ and $n_{s_0}^x = n_s^x$, let $n = n_s^x - 2$, and set

$$\begin{array}{rcl} m_2(x,s) & = & \sum_{n' < n_s^x - 2}((n'+1)^2 + 2) + m_2(s) \\ m_1(x,s) & = & m_1(s) \\ m_0(x,s) & = & m_0(s) \end{array}$$

where $m_2(s), m_1(s)$ and $m_0(s)$ are defined as in Claim 5 for the given parameters n and s_0. Finally, for $s < s^*$, let $m_j(x,s) = m_j(x,s^*)$.

Note that, for $s \geqslant s^*$ and for n and s_0 as above, there is no stage $s'+1$ such that $s_0 < s'+1 \leqslant s$ and such that a requirement $\mathcal{R}_{n'}$ with $n' \leqslant n$ becomes injured at stage $s'+1$ since otherwise, by choice of s^*, the requirement $\mathcal{R}_{2\langle e,x\rangle+i}$ would be linked to $\mathcal{R}_{n_s^x-2}$ at stage $s'+1$ hence $n_{s'+1}^x < n_s^x$ contrary to $s'+1 \leqslant s$. So, by Claim 5 (i), the definition of $m_j(x,s)$ is sound for $s \geqslant s^*$ (and so for $s \geqslant 0$ in general), and, obviously, the functions $m_j(x,s)$ are computable. So it only remains to verify (14) and (15). Fix x where w.l.o.g. $x \geqslant 2$ and $s \geqslant 0$. If $s < s^*$ then $m_j(x,s) = m_j(x,s+1)$ for $j \leqslant 2$ and $\psi(x,s) = \psi(x,s+1)$. So w.l.o.g. we may assume that $s \geqslant s^*$. Let $n = n_s^x$ and $s_0 = \mu\, s' \leqslant s\ [s' \geqslant s^* \ \&\ n_{s'}^x = n_s^x]$.

Now, for a proof of (14), note that (14) holds by Claim 5 (ii) if $n^x_s = n^x_{s+1}$. So it suffices to show

$$\begin{gathered} n^x_{s+1} < n^x_s \;\Rightarrow \\ (m_2(x,s+1), m_1(x,s+1), m_0(x,s+1)) < (m_2(x,s), m_1(x,s), m_0(x,s)). \end{gathered} \tag{17}$$

(Actually, for the proof of (14) it suffices to have $\leqslant$ in the conclusion; but we will need this stronger fact for the proof of (15) below.) So assume $n^x_{s+1} < n^x_s$. Then

$$\begin{array}{rcl} m_2(x,s) & \geqslant & \sum_{n' < n^x_s - 2}((n'+1)^2 + 2) \\ & & \text{(by definition of } m_2(x,s)) \\ & > & \left(\sum_{n' < n^{x+1}_s - 2}((n'+1)^2 + 2)\right) + (n^x_{s+1} - 2 + 1)^2 + 1 \\ & & \text{(by } n^x_{s+1} < n^x_s) \end{array}$$

while, by definition of $m_2(x,s+1)$,

$$m_2(x,s+1) = \left(\sum_{n' < n^{x+1}_s - 2} ((n'+1)^2 + 2)\right) + m_2(s+1)$$

where $m_2(s+1) \leqslant (n^x_{s+1} - 2 + 1)^2 + 1$. Hence $m_2(x,s) > m_2(x,s+1)$ which completes the proof of (17).

Finally, for a proof of (15), w.l.o.g. we may assume that $\psi(x,s+1) \neq \psi(x,s)$. It suffices to show

$$(m_2(x,s+1), m_1(x,s+1), m_0(x,s+1)) < (m_2(x,s), m_1(x,s), m_0(x,s)). \tag{18}$$

holds.

Since $x \geqslant 2$ and $\psi(x,s+1) \neq \psi(x,s)$ it follows by definition of ψ that $\psi(x,s) = \Phi^{A_{i,s}}_{e,s}(x)\downarrow$, $s \geqslant \max\{2\langle e,x\rangle + i + 1, s^*\}$ and (16) holds whereas $\Phi^{A_{i,s+1}}_{e,s+1}(x) \neq \Phi^{A_{i,s}}_{e,s}(x)$ or (16) fails for $s+1$ in place of s. Since, by (17), w.l.o.g. we may assume that $n^x_{s+1} = n^x_s$, it follows that a number $< \varphi^{A_{i,s}}_{e,s}(x) \leqslant r(n^x_s, s)$ entered A_i at stage $s+1$ or $r(n^x_s, s+1) < r_e(n^x_s, s)$. But either implies that requirement $\mathcal{R}_{n^x_s}$ is injured at stage $s+1$. So, by Claim 5 (iii), (12) holds. By definition of $m_j(x,s)$, this implies (18).

This completes the proof of Claim 6.

Claim 7. A_0 and A_1 are low.

Proof. Given $i \leqslant 1$, in order to show that A_i is low it suffices to show that the lowness requirements

$$\mathcal{Q}_{2e+i}:\ \exists^\infty s(\Phi^{A_{i,s}}_{e,s}(e)\downarrow) \;\Rightarrow \Phi^{A_i}_e(e)\downarrow$$

are met (for $e \geqslant 0$). So fix e such that $\Phi^{A_{i,s}}_{e,s}(e)$ is defined for infinitely many s. By Claim 2 choose a stage s_0 such that the requirement $\mathcal{R}_{2\langle e,e\rangle + i}$ is not injured after stage s_0, and by choice of e fix $s > s_0$ such that $\Phi^{A_{i,s}}_{e,s}(e)$ is defined. Then

$$\varphi^{A_{i,s}}_e(e) = r_{\langle e,e\rangle}(2\langle e,e\rangle + i, s) \leqslant r(2\langle e,e\rangle + i, s) \leqslant R(2\langle e,e\rangle + i, s).$$

Moreover, since $\mathcal{R}_{2\langle e,e\rangle + i}$ is not injured after stage s_0,

$$A_i \restriction R(2\langle e,e\rangle + i, s) = A_{i,s} \restriction R(2\langle e,e\rangle + i, s).$$

So the computation $\Phi^{A_{i,s}}_{e,s}(e)$ is preserved, i.e., $\Phi^{A_i}_e(e) = \Phi^{A_{i,s}}_{e,s}(e)\downarrow$.

This completes the proof of Claim 7 and the proof of Theorem 4.3. □

We remark that Sacks splitting theorem has a stronger form: Given noncomputable c.e. C (even Δ_2^0), then there exist a c.e. splitting $A_1 \sqcup A_2 = A$ such that $C \nleqslant_T A_i$, for $i \in \{1, 2\}$.

Theorem 4.4 (Ambos-Spies, Downey, Monath, Ng [2])**.** For any $\alpha < \epsilon_0$, there are noncomputable c.e. C and A such that if $A_1 \sqcup A_2 = A$ is a c.e. splitting of A and both A_1 and A_2 are α-c.a. then $C \leqslant_T A_1$ or $C \leqslant_T A_2$.

That is, Sacks splitting with cone avoidance really is finite injury of unbounded type. In fact the theorem applies to degree splits.

5. Maximal α-c.a. degrees

A remarkable phenomenom is that the new hierarchy gives new definable antichians based on *maximality*.

Definition 5.1. We say that $\mathbf{a}$ has maximal totally α-c.a. degree if

- $\mathbf{a}$ is totally α-c.a., and
- For all $\mathbf{b} > \mathbf{a}$, $\mathbf{b}$ is not totally α-c.a.

Cholak, Downey and Walk [15] constructed maximal *contiguous* degrees. With easier constructions, Downey and Greenberg established the following.

Theorem 5.2 ([22])**.**

(1) If $\alpha < \epsilon_0$ is a power of ω, then there exists a maximal totally α-c.a. degree.
(2) In fact, if $\alpha < \epsilon_0$ is a power of ω, then there exists a maximal totally α-c.a. degree, which is also uniformly totally α-c.a. degree.
(3) On the other hand, maximality has its limits. For example, if $\beta < \epsilon_0$, then every totally ω^β-c.a. degree lies (strictly) below a totally $\omega^{\beta+1}$-c.a. degree. Thus no totally ω^β-c.a. degree can be maximal totally $\omega^{\beta+1}$-c.a.
(4) If $\alpha < \epsilon_0$ is a limit of powers of ω then no c.e. degree is maximal totally $< \alpha$-c.a. In particular, there are no maximal $< \omega^\omega$-c.a. degrees.

Corollary 5.3 ([22])**.**

(1) There is a definable antichain in the c.e. degrees given by the maximal totally ω-c.a. degreees; namely those that do not bound critical triples but every degree above them does.
(2) There is no maximal degree which does not bound M_5.

Recent work of the first two authors together with Katherine Arthur [7, 6] has explored the relationship of maximal α-c.a. degrees and the rest of the hierarchy.

Theorem 5.4 (Arthur, Downey and Greenberg [6, 7])**.**

(1) Let $\alpha < \epsilon_0$ be a power of ω, and let $\mathbf{a}$ be a totally α-c.a. degree. Suppose that $\beta > \alpha^\omega$ also a power of ω. Then there is a $\mathbf{b} > \mathbf{a}$ which is maximal β-c.a.
(2) Suppose that $\mathbf{a}$ is totally ω-c.a. Then there is a $\mathbf{b} > \mathbf{a}$ which is totally ω^4-c.a. and not totally ω-c.a.
(3) Suppose that $\mathbf{a}$ is superlow. Then there exists $\mathbf{b} > \mathbf{a}$ which is maximal totally ω-c.a.
(4) There is a totally ω-c.a. degree $\mathbf{a}$ which is not bounded by any maximal totally ω-c.a. degree.

Theorem 5.4 (2) was improved by Li Ling Ko, who gave a nonuniform proof based on work from [22, 7] that **b** could be taken as ω^2. We also have a kind of minimal cover.

Theorem 5.5 (Arthur, Downey and Greenberg [6, 7])**.** *There are c.e. degrees* $\mathbf{a} < \mathbf{b}$ *such that* $\mathbf{b}$ *is totally* ω*-c.a. and such that if* $\mathbf{c} > \mathbf{a}$ *is totally* ω*-c.a., then* $\mathbf{c} \leqslant \mathbf{b}$*.*

One of the primary open questions here is whether every totally ω-c.a. degree is bounded by a totally ω^2-c.a. degree, or a totally ω^n-c.a. degree for some $n > 2$; in the other extreme, perhaps there is a single totally ω-c.a. degree **a** which is not bounded by a maximal totally ω^n-c.a. degree for any n. Currently we conjecture that the answer is no. The authors proved the following.

Theorem 5.6. *There is no uniform way to find a maximal totally* ω^2*-c.a. degree above a given totally* ω*-c.a. degree.*

We put the very complex proof from Hammatt's MSc Thesis in the last section of this paper for the sake of narrative flow. The proof is a $\mathbf{0}'''$-priority argument, and shows the depth of the material.

6. A New Hierarchy

The ideas of the Downey-Greenberg hirarchy have been used to construct a new hierarchy based on the *wtt*-jump.

We know that Post suggested "thinness" properties of complements of sets might solve his problem of finding a Turing incomplete c.e. set. We know that in its original form this proposal fails, since

1. As we have seen, Martin [47] proved there are complete maximal c.e. sets (recall M is maximal if it is a co-atom in $\mathcal{L}^*$), and indeed every high c.e. degree has a maximal set.
2. Soare [62] showed that all maximal sets are automorphic so no "extra" property will suffice to guarantee incompleteness.
3. Furthermore, Cholak, Downey, Stob [14] extablished that no property of the complement of a c.e. set alone can guarantee incompleteness.

As we mentioned earlier, however, Harrington and Soare [37] showed that there is a definable property Q such that if $Q(A)$ then A is incomplete.

On the other hand, there are fascinating interactions with strong reducibilities:

- Simple sets solve Post's problem for m-degrees (Post [57]);
- (η-) Maximal sets have minimal m-degrees (Ershov [31], Lachlan [42]);
- Simple sets are not btt-cuppable (Downey, [18]);
- Dense simple sets are not tt-cuppable (Kummer and Schaefer, [39]);
- Hypersimple sets are wtt-incomplete and indeed not wtt-cuppable (Downey and Jockusch, [20]).

In Theorem 3.8, we saw that Barmpalias, Downey and Greenberg proved that a c.e. **a** is is totally ω-c.a. iff every (c.e.) set in **a** is wtt-reducible to a h-simple c.e. set.

Inspired by this result, Ambos-Spies, Downey and Monath attacked the question of trying to characterize c.e. sets wtt-reducible to maximal sets. The answer turned out to yield a fascinating new hierarchy. The preliminary result was the following:

Recall that A is *superlow* if $A' \equiv_{tt} \varnothing'$. Equivalently, for c.e. A, there is a computable h such that $J^A(e)$ is h-c.a. Here J^A is the universal partial A-computable function.

Theorem 6.1 (Ambos-Spies, Downey, and Monath)**.** If A is c.e. and superlow, then there is a maximal set M with $A \leqslant_{wtt} M$. Indeed $A \leqslant_{ibT} M$.

Proof. (sketch) We build $A \leqslant_{wtt} M$ meeting.

$$R_e : W_e \cap \overline{M} \text{ infinite implies } W_e \supseteq^* \overline{M}.$$

$$N_e : \lim_s m_{e,s} = m_e \text{ exists where } m_{0,s} < m_{1,s} \dots \text{ lists } \overline{M_s}.$$

A standard maximal set construction maximizes *e-states.* The e-state of $z \in \overline{M_s}$ is $\{j \leqslant e \mid z \in W_{j,s}\}$, a string. The construction tries to put almost all of $\overline{M}$ into the same e-state. If $\Gamma^M = A$ is the wtt-reduction, then if some $x \in A_{s+1} - A_s$, we need to change $M\upharpoonright_\gamma (x)$. This can only be done if there is some element in $\overline{M_s}$ *which is below* $\gamma(x)$ which can be put into $M - M_s$. We refer to this as *coding.* The e-state machinery puts lots of elements from $\overline{M_t}$ into M_{t+1}, so we must be careful to leave *enough* elements to cope with this coding. For example, even for the 0-state, i.e. for a single requirement, all of the elements in $W_{0,s} \cap \overline{M_s}$ might be bigger than $\gamma(0)$ so we could not code 0 entering A.

We only do this e-state action when we can use the jump computation to tell us that we are safe, and few elements will enter A.

For One requirement R_0 We have some part of the jump we control using the recursion theorem say $J^A(\langle g(0), j\rangle \mid j \in \mathbb{N}\}$. Jump computations on this have an approximation (known in advance) $J^A(\langle g(0), j\rangle)[s]$ changing at most $h(\langle g(0), j\rangle)$ many times. Anticipating things somewhat we write that as $f_0^A(j)[s]$ and the mind change number $n(0, j)$.

For a single requirement, set aside a block of elements B_1 with at least $n(0,0)+1$ many elements which we *don't* raise the 0-state of. When we see at least $n(0,1)+1$ many elements ($> \max B_1$) in the high state, the plan is to use these for B_2 and we define $f_0^A(0)[s]\downarrow$, with huge use s, and wait for the approximation to be confirmed. We define the interval $I_1 = [\max B_1, s]$, and . We no longer code below $\max B_1$.

Now we declare that B_1 will code I_1, and dump all elements not in $B_1 \sqcup B_2$ below $s_0 = s$ into $M_{s+1} - M_s$.

Each time some element enters A between $\max B_1$ and $\max I_1$ we redefine the jump on argument 0 with use s_0 and on recovery we code all such this using an element in B_1.

We repeat the process planning to use B_2 to code some interval $[\max I_1, s_1] = I_2$.

Now the coding is in the high state That is, we's wait for at least $n(0,2)+1$ many elements in the high state (and these *must* be $> \max I_1$) for block B_3, etc. So block B_n looks after I_n.

For more than one states, this is done inductively. First note that B_1 might never be used as maybe the 0-state is not well-resided in $\overline{M}$. So there would need to be a version of "B_1 for R_1 *guessing* R_0 is inactive" and working in the same way as above for R_0, and getting re-stated each time the 0 state acts up.

There would be a version of R_1 guessing R_0 is infinitely often active. This demands that B_2 above would have a part of its block devoted to $B_1^{\infty\infty}$. It is only used when we see enough elements in state $\infty\infty$ and these are verified by a part of the jump we build for this guess $f_{\infty\infty}^A(1)$. □

The key insight is that all of the definitions above are *wtt-jump* computations, in that the use *never changes once defined.* That is, the proof only needs the following new concept:

Definition 6.2. We say that A is *wtt-sl* (*wtt-superlow*) iff $\hat{J}^A$ is ω-c.a., where $\hat{J}$ is the partial wtt-jump.

Here to define the wtt-jump, we can list all wtt-precedures via pairs (Φ_e, φ_e). And then allow $\Phi_e^X(e)\downarrow$ if the use is bounded by φ_e. Note that saying something is wtt-sl is to say that the value of the the wtt-jump relative to A is $\leqslant_{wtt} \varnothing'$. The proof above gives the followling.

Theorem 6.3 (Ambos-Spies, Downey and Monath)**.** $A \leqslant_{ibT} M$ for M maximal if A is c.e. and wtt-sl.

The wtt-jump is quite different from the Turing one. For example we have the following:

Theorem 6.4 (Ambos-Spies, Downey and Monath)**.** There are wtt-sl Turing complete c.e. sets.

In fact, Ambos-Spies, Downey and Monath (in preparation) have shown that if we define A to be *wtt strongly superlow* to mean that the wtt jump is wtt-reducible to $\varnothing'$ with arbitarily slow growing use, then the Turing degree of $\varnothing'$ contains wtt-strongly superlow c.e. sets.

6.1. **The characterization.** A modified version works for the following.

Definition 6.5. A is *eventually uniformly wtt-array computable* iff there are computable functions k, g and h, with $k(n,s) \leqslant k(n,s+1) \leqslant 1$, $\lim_s k(n,s)$ exists for all n such that

(1) $\lim_s g(x,s) = \hat{J}^A(x)$ for all x.
(2) $k(n,s) \leqslant k(n,s+1) \leqslant 1$, $\lim_s k(n,s)$ exists for all n
(3) If $k(x,s) = 1$ then $g(x,t)$ has at most $h(x)$ further mind changes for $t > s$ (hence wlog $k(x,t) = 1$ for all $t > s$).
(4) If $\hat{J}^A(\langle e,y\rangle)\downarrow$ for all y, then for almost all s, $\lim_s k(\langle e,y\rangle, s) = 1$.

More or less the same proof gives one direction of:

Theorem 6.6 (Ambos-Spies, Downey and Monath)**.** For a c.e. A, $A \leqslant_{ibT} M$ iff $A \leqslant_{wtt} M$ for M maximal iff A is eventually uniformly *wtt*-array computable.

Proof. We also sketch a proof of the other direction.

- Suppose that $\Gamma^M = A$, and A not eventually uniformly wtt-ac. Choose $h(n) = 2^n$ for simplicity.
- Let $\ell(s) = \max\{z \mid\leqslant y\Gamma^M\restriction_z = A\restriction_z [s]\}$.

Our assumptions about the enumerations of A, M and the jump are that once $\ell(s) > n$, if $A_{s+1}(n) \neq A_s(n)$, $M_{s+1}\restriction_\gamma (n) \neq M_s\restriction_\gamma (n)$. We know that for any approximation for the wtt-jump $g(\langle e,x\rangle, s)$ there will be total $\hat{\Phi}_e^A(x)$ changing more than $h(\langle e,x\rangle)$ many times on infinitely many x. Initially we can have $k(\langle e,x\rangle, s) = 0$ for all e, x and keep it like this unless told otherwise for $t > s$. The approximation $g(\langle e,x\rangle, s)$ is the natural one observing halting computations.

For each e carry out the following construction. When we see $\hat{\Phi}_e^A(0)\downarrow [s]$ let $I_0^e = [0, \gamma(\phi_e(0)))$. Without loss of generality, the $\hat{\Phi}_e^A$'s are monotone, and we can continue $I^e_1 = [\gamma(\phi_e(0)), \gamma(\phi_e(1)))$, defining a sequence of disjoint e-intervals $\{I^e_n \mid n \in \mathbb{N}\}$. If ever we see $|\overline{M}_s \cap [0, \max I^e_n)| < 2^{\langle e,n\rangle}$, define $k(\langle e, n\rangle, s) = 1$. (Note that this ensures $k(\langle e, n\rangle, t) = 1$ for all $t > s$, also).

The assumption is that $\Gamma^M = A$, and M is maximal. Notice that if we define $k(\langle e, n\rangle, s) = 1$, $A\upharpoonright_{\phi_e(n)}$ can change only $< 2^{\langle e,n\rangle} = h(\langle e, n\rangle)$ many times, since each change induces a change in $M\upharpoonright_\gamma (\phi_e(n))$ and hence $M\upharpoonright_{\max} I^e_n$. There are not enough elements to enter $M - M_s$ for this to happen more than $2^{\langle e,n\rangle} - 1$ many times. So 3 of Definition 6.5 holds.

Now suppose that $\hat{\Phi}^A_e$ is total. Then for each n, we define I^e_n. Moreover, since M is maximal, we know that for almost all n, $|\overline{M} \cap I^n_e| \leqslant 1$. Thus, for almost all n, there is an s with $|\overline{M}_s \cap [0, \max I^e_n)| < 2^n \leqslant 2^{\langle e,n\rangle}$, and hence for all e with $\hat{\Phi_e}^A$ total, and for almost all n, s, $k(\langle e, n\rangle, s) = 1$.

Therefore A is eventually uniformly wtt-ac, a contradiction. □

We remark that the same argument works for other classes of c.e. sets in place of maximal sets including dense simple, and hh-simple sets.

This result says that we should further work towards understanding approximations to wtt-functionals.

Ambos-Spies, Downey and Monath have some modest progress in this area.

Theorem 6.7 (Ambos-Spies, Downey and Monath)**.** *Let $A \leqslant_{wtt} B$ be any sets. Then if B is eventually uniformly wtt-array computable, so is A.*

Interestingly, for c.e. sets, the class is closed under join:

Theorem 6.8 (Ambos-Spies, Downey and Monath)**.** *If A and B are eventually uniformly wtt-array computable c.e. sets then so is $A \oplus B$.*

Hence, the wtt-degrees of eventually uniformly wtt-array computable c.e. sets form an ideal in the wtt degrees.

We can also explore connections between wtt-jump traceability and wtt-superlowness, depending on the growth rate of orders, in the same was as we have seen for the jump tracinng hierarchy. That is, classically we would say that A is h-jump traceable iff there is a computable collection of c.e. sets $\{W_{f(n)} \mid n \in \omega\}$ and $|W_{f(n)}| \leqslant h(n)$ such that if $J^A(n)\downarrow$, then $J^A(n) \in W_{f(n)}$. Then A is strongly jump traceable if A is jump traceable for all orders h. This is a fascinating class of reals associated with algorithmic randomness (see [24, 36]). We can pursue the same for $\hat{J}$ in place of J. Some preliminary results and other results about analogs of strong jump traceability can be found in Monath's Thesis [51].

7. Proof of Theorem 5.6

Now we detail the construction to prove Theorem 5.6. The result will follow from an application of the Recursion Theorem to the argument below.

We build a set A for a given W such that $\deg_T(A)$ is totally ω-c.a. and $\deg_T(A \oplus W)$ is not maximal totally ω^2-c.a. To ensure A has the desired properties we meet the following set of requirements:

- N_Φ: If $\Phi(A)$ is total then $\Phi(A)$ is ω-c.a.
- P_Ψ: If $\Psi(A, W, Q)$ is total then either it is ω^2-c.a. or $\Gamma_\Psi(A, W)$ is total and not ω^2-c.a. Subrequirements:

  - $P_{\Psi,k}$: $\Gamma_\Psi(A,W) \neq f^k$ where f^k is the kth ω^2 c.a. function, along with its approximation $\langle f^k_s, o^k_s\rangle$
- R_Θ: $\Theta(A,W) \neq Q$ or $\Delta_\Theta(A,W)$ is not ω^2-c.a. Subrequirements:
  - $R_{\Theta,k}$: $\Delta_\Theta(A,W) \neq f^k$ where f^k is the kth ω^2 c.a. function, along with its approximation $\langle f^k_s, o^k_s\rangle$

7.1. Glossary.

Here we give a glossary of terms that will be used in this chapter as well as the following chapter. These terms will be introduced during the technical discussion.

- π is a node working for an N_Φ requirement.
- τ is a node working for a P_Ψ requirement. A *daughter* of τ works for requirement $P_{\Psi,k}$ for some $k < \omega$. A *son* of τ is a ζ_k node for some $k < \omega$.
- ρ is a node working for a $P_{\Psi,k}$ requirement. τ is the *parent* of ρ is τ works for requirement P_Ψ.
- η is a node working for an R_Θ requirement. A *daughter* of η works for requirement $R_{\Theta,k}$ for some $k < \omega$. A *son* of η is an ξ_k node for some $k < \omega$.
- μ is a node working for an $R_{\Theta,k}$ requirement. η is the *parent* of μ is η works for requirement R_Θ.
- $\mathtt{tr}_s(\rho, x)$ is the tracker for (ρ, x) at stage s. We sometimes use the notation $\mathtt{tr}_s(x)$ when it is clear which ρ this refers to.
- $\mathtt{ac}_s(\mu)$ is the anchor for μ at stage s and $\mathtt{fl}_s(\mu)$ is the follower for μ at stage s.
- $\mathtt{pro}_s(\mu)$ is a set of (ρ, x) that μ *protects*.
- $\mathtt{pro}_s(\hat\rho, \hat x)$ is a set of (ρ, x) that $(\hat\rho, \hat x)$ *protects*.
- $\langle f^\alpha_s, o^\alpha_s\rangle_{s<\omega}$ is the ordinal approximation given by an α node, where α is either a μ or ρ node. $o^\rho_s(z)$ has the form $o^\rho_s(z) = \omega \cdot d^\rho_s(z) + b^\rho_s(z)$.
- $\Gamma_\Psi(A,W) = \Gamma_\tau(A,W) = \Gamma_\rho(A,W)$ and $\Psi_\tau(A,W,Q) = \Psi_\rho(A,W,Q)$ if τ works for requirement P_Ψ and ρ is a daughter of τ. Similarly $\Delta_\Theta(A,W) = \Delta_\eta(A,W) = \Delta_\mu(A,W)$ and $\Theta_\eta(A,W) = \Theta_\mu(A,W)$ if η works for requirement R_Θ and μ is a daughter of η.
- $I_s(\rho, x)$ is a set of elements from ω such that x is the least element in the set and every other $x' \in I_s(\rho, x)$ has been declared taken over by x at some stage $t < s$.
- $\mathcal{C}_s(\rho)$ is the set of inputs x that have been established and not taken over for ρ. $\mathcal{C}_0(\rho) = \varnothing$ and then at each stage $\rho\hat{}\infty$ is accessible, a new input x is established. As inputs are taken over we remove them from $\mathcal{C}_s(\rho)$ so this set contains the least input from every interval for ρ.
- When (ρ, x) is *invented* means the first tracker for x has been appointed, we call this tracker the *original* tracker for x, denoted $\mathtt{orig}(x)$.
- (ρ, x) is *established* once there has been a ρ-expansionary stage since the invention of x.
- We call (ρ, x) *corrupted* while the tracker of x is not the original tracker of x. This happens when a μ node enumerates a number into Q.
- (ρ, x) *fully corrupted* means there has been a ρ-expansionary stage after the corruption of x, this means the ordinal for the new tracker has been revealed.
- If (ρ, x) is *uncorrupted* then the new tracker, that was appointed after corruption, has been replaced by the original tracker. Then happens when

there is a relatively small A or W change that allows us to use the orginal tracker after corruption.

- Suppose x is corrupted. Then the *corrupting* μ is the μ node that enumerated a number into Q at the stage x is declared corrupted. The *corrupting* q is the number that the corrupting μ enumerated into Q.
- We declare (ρ, x) to start an attack at the start of the stage. We say (ρ, x) is *fully in an attack* once there has been a τ-expansionary stage during the attack. Note that $\Gamma_\tau(A, W, \mathtt{tr}(\rho, x)) \uparrow$ while (ρ, x) is in an attack but not yet fully in an attack.
- $\psi^\rho_s(I_s(\rho, x)) = \max\{\psi^\rho_s(x) : x \in I_s(\rho, x)\}$
- Define $\#t$ to be the largest number used by the construction at stage t.
- $\mathtt{prec}(\alpha)$ is the set of τ such that there is $\zeta_k \uparrow \preceq \alpha$ where ζ_k is a son of τ.
- A node μ believes $\Psi_\tau(A, W, Q)$ is total if $\mu > \rho\hat{}\infty$ where ρ is a daughter of τ.
- A node μ believes $\operatorname{dom} \Psi_\tau(A, W, Q) = k$ if $\mu > \zeta_k \uparrow$ where ζ_k is a son of τ.
- When τ carries out a request or passes a request on to some $\hat{\tau}$ we say that it *acts on the request.*
- (ρ, x) *requires protection from* μ *at stage* s if $x \in \mathcal{C}_s(\rho)$, (ρ, x) is not currently in an attack and one of the following holds:
  - $\rho\hat{}\infty \preceq \mu$
  - ρ is to the left of μ and μ believes $\Psi_\rho(A, W, Q)$ is total
  - ρ is to the left of μ and μ believes $\operatorname{dom} \Psi_\rho(A, W, Q) = k$ and $I_s(\rho, x) \subseteq k$

7.2. **Technical Discussion.** First consider the requirement R_Θ. If $\Theta(A, W)$ is not total then we have met the requirement, otherwise we build a functional $\Delta_\Theta(A, W)$. If $\Theta(A, W)$ is total and $\Delta_\Theta(A, W)$ is ω^2-c.a. then we need to enumerate a set Q such that $\Theta(A, W) \neq Q$. A node on the tree of strategies working for requirement R_Θ is denoted η. η guesses whether $\Theta(A, W)$ is total by looking at the length of $\operatorname{dom} \Theta(A, W)$. η nodes have two outcomes, one that believes $\operatorname{dom} \Theta(A, W)$ goes to infinity and the other believes $\operatorname{dom} \Theta(A, W)$ is finite. Below the finite outcome we do not need to act for this requirement as if this is the correct outcome then we have met the requirement. At stages that we believe the infinite outcome we need to build an initial segment of the functional $\Delta_\Theta(A, W)$. Below the infinite outcome we need to guess if $\Delta_\Theta(A, W)$ is ω^2-c.a. This gives subrequirements $R_{\Theta,k}$ which guess whether $\Delta_\Theta(A, W) = f^k$ or not, where f^k is the kth ω^2 c.a. function. The nodes working for subrequirements are called *children* of η and are denoted μ. μ nodes have two outcomes, one that believes $\Delta_\Theta(A, W) = f^k$ and the other believes $\Delta_\Theta(A, W) \neq f^k$. Now notice that children of η are only on the tree below the infinite outcome of η. Also notice that below the outcome of child node that believes $\Delta_\Theta(A, W) = f^k$ we do not need to place any more children of η. Now at stages where we believe that $\Delta_\Theta(A, W)$ is ω^2-c.a., which is indicated by the infinite outcome of a μ node, we need to ensure that $\Theta(A, W) \neq Q$. To diagonalise we appoint a follower q and then at stages s where we see $\Theta(A, W, q)[s] = 0$, we enumerate q into Q. Now suppose we enumerated q into Q at stage s because $\Theta(A, W, q)[s] = 0$. Notice that if there is an A or W change below the use $\theta_s(q)$ after stage s, the computation $\Theta(A, W, q)[s]$ is injured, so it is possible that $\Theta(A, W, q) = 1$; hence our diagonalisation has failed. If this happens we appoint a new follower and make another attempt to diagonalise. Now to guarantee that diagonalisation is successful we need to ensure that only finitely many followers are appointed. To do this we appoint an anchor p which will serve many followers and use the fact that $\Delta_\Theta(A, W)$ is ω^2-c.a. So when we define the computation $\Delta_\Theta(A, W, p)$ we define it with use at least $\theta(q)$, where q is the current follower. Now when we enumerate q into Q because we have seen $\Theta(A, W, q)[s] = 0$, if here is an A or W change below the use of this computation then it is also below the use of the computation $\Delta_\Theta(A, W, p)$. Now since $\Delta_\Theta(A, W)$ is ω^2-c.a., the ordinal of the ω^2-computable approximation must decrease. This can only happen finitely many times; hence only finitely many followers are appointed and the last follower must be successful.

Now consider the requirement P_Ψ. Similar to the R requirements, if $\Psi(A, W, Q)$ is not total then we have met the requirement, otherwise we build a functional $\Gamma_\Psi(A, W)$. Then if $\Psi(A, W, Q)$ is total and $\Gamma_\Psi(A, W)$ is ω^2-c.a. then we need to show that $\Psi(A, W, Q)$ is also ω^2-c.a. A node on the tree of strategies working for requirement P_Ψ is denoted τ. A τ node guesses whether $\Psi(A, W, Q)$ is total or not by looking at the length of $\operatorname{dom} \Psi(A, W, Q)$. These τ nodes work in the same way as the η nodes. At stages that we believe the infinite outcome we need to build an initial segment of the functional $\Gamma_\Psi(A, W)$. Then below the infinite outcome we need to guess if $\Gamma_\Psi(A, W)$ is ω^2-c.a. So similar to the R requirements, we get subrequirements guessing whether $\Gamma_\Psi(A, W) = f^k$ or not. Nodes working for these subrequirements are called *children* of τ and are denoted ρ. ρ nodes have two outcomes similar to μ nodes. If $\Gamma_\Psi(A, W)$ is ω^2-c.a. then there is a ρ node which guesses $\Gamma_\Psi(A, W) = f^k$. Then this ρ node gives us an ω^2-computable approximation $\langle f^\rho_s, o^\rho_s \rangle$. So for each $z < \omega$ there is an ordinal $o^\rho_s(z)$ with the form $\omega \cdot d^\rho_s(z) + b^\rho_s(z)$. Now to show $\Psi(A, W, Q)$ is ω^2-c.a. we need to define an ω^2-computable approximation for $\Psi(A, W, Q)$. For each $x < \omega$ we need to define a non-increasing sequence of ordinals $o^\Psi_s(x)$ of the form $\omega \cdot m^\Psi_s(x) + k^\Psi_s(x)$ such that if $\Psi(A, W, Q, x)[s + 1] \neq \Psi(A, W, Q, x)[s]$ then $o^\Psi_{s+1}(x) < o^\Psi_s(x)$. To do this we appoint a tracker for each $x < \omega$, $\mathtt{tr}(x)$, once we have appointed a tracker for x we say x has been *invented*. When we first issue an ordinal to x we say x has been *established*. Fix x and let z be the tracker of x. First assume Q is empty. When we define the computation $\Gamma_\Psi(A, W, z)$ at stage s we define it with use at least $\psi_s(x)$. Now if $\Psi(A, W, Q, x)[s + 1] \neq \Psi(A, W, Q, x)[s]$, then there has been some change below the use $\psi_s(x)$. Since $\gamma_s(z) \geqslant \psi_s(x)$ and Q is empty, it follows that $\Gamma_\Psi(A, W, z)[s + 1] \neq \Gamma_\Psi(A, W, z)[s]$; hence $o^\rho_{s+1}(z) < o^\rho_s(z)$. Then we see that we can just follow the ordinal of the tracker. So if we define $m^\Psi_s(x) = d^\rho_s(z)$ and $k^\Psi_s(x) = b^\rho_s(z)$ then this gives us an ω^2-computable approximation for $\Psi(A, W, Q)$.

So now we need to consider how R requirements enumerating numbers into Q affect the P requirements. Fix x and let z be the tracker of x. Suppose at stage s an R requirement enumerates its follower q into Q and $q < \psi_s(x)$. Now $\Psi(A, W, Q, x)[s+1] \neq \Psi(A, W, Q, x)[s]$ but it could be that $\Gamma_\Psi(A, W, z)[s+1] = \Gamma_\Psi(A, W, z)[s]$ and $\psi_{s+1}(x) > \gamma_{s+1}(z)$. Then the current tracker is useless, so we need to appoint a new tracker z' and we declare that x is *corrupted*. Now this new tracker has an ordinal $o^\rho_s(z') = \omega \cdot d^\rho_s(z') + b^\rho_s(z')$. But it is quite possible the $d^\rho_s(z') > d^\rho_s(z)$, so we cannot just follow the ordinal of the new tracker like our previous strategy. Instead, when x gets corrupted we decrease $m^\Psi(x)$ by one and this allows us to define a new large value for $k^\Psi(x)$, so we define $k^\Psi(x) = d^\rho(z')$. Now consider what happens if there is a W change below $\psi(x)$. Now $o^\rho(z')$ must have decreased since this change was also below $\gamma(z')$ but it could be that $d^\rho_s(z')$ has not changed.

But now $k^{\Psi}(x) = d^{\rho}(z')$ and we need to see a decrease in $k^{\Psi}(x)$. So we begin an *attack* where we lift the use $\gamma(z')$ to be large and enumerate this into A. Each time we do this we will see the ordinal $o^{\rho}(z')$ decrease, so eventually we will see $d^{\rho}(z')$ decrease and this gives us the decrease in $k^{\Psi}(x)$ that we needed.

Now suppose x has been corrupted and let $\texttt{orig}(x)$ be the original tracker for x. If there is a stage t such that $\Gamma_{\Psi}(A, W, \texttt{orig}(x))[t] \uparrow$ then we are able to once again define the use of this computation to be at least the use $\psi(x)$; hence we can use $\texttt{orig}(x)$ as the tracker for x once again. If this happens then we call x *uncorrupted.* Once x is uncorrupted then we decrease $m^{\Psi}(x)$ by one and define $k^{\Psi}(x)$ to be $b^{\rho}(\texttt{orig}(x))$ and go back to following the ordinal of the original tracker as discussed while Q was empty. We can continue to do this until there is another stage where a number enters Q below $\psi(x)$.

To meet the N requirements we first need to guess whether $\Phi(A)$ is total or not, we do this in the same way as the P and R requirements. If $\Phi(A)$ is not total then we are done with the requirement, otherwise we need to show that $\Phi(A)$ is ω-c.a. First consider the interaction of only one P requirement with an N requirement. Now when y is established, at stage s, we can see the ordinal, $o_s(\texttt{tr}(x)) = \omega \cdot d_s(\texttt{tr}(x)) + b_s(\texttt{tr}(x))$, for established inputs x. Now if x has already been corrupted then if there is a W change an attack could start and this attack puts at most $b_s(\texttt{tr}(x))$ numbers into A. Since this number has been seen by y when it was established we can allow injury from this attack. Notice that if we did let an attack beat the ordinal all the way down y has no idea how many injuries this could cause because at stage s all it can incorporate into its ordinal is $b_s(\texttt{tr}(x))$; hence we can see that is important that we stop attacking once $d(z)$ has decreased by one. Once this attack finishes we cannot let any future (ρ, x) attack injure $\Phi(A, y)$ because we had not seen the bound for any future (ρ, x) attacks when y was established.

Consider the interaction between two P requirements. We only start an attack when there is a W change; hence we need to be able to control how one P requirement could cause an A enumeration that could injure the computation of another P requirement. Now recall that since we are done if $A \oplus W$ is not totally ω^2-c.a. which allows us to be able avoid having another parent node on the tree between a parent and its child. Hence if $\hat{\rho} \geq \rho\hat{\ }\infty$ then $\hat{\tau} \geq \rho\hat{\ }\infty$. This is very useful because this way $\hat{\tau}$ can see that $\Gamma(A, W)$ is totally ω^2-c.a. So during an attack it can define a set of inputs that it must *protect*, denote this set by $\texttt{pro}(\hat{\rho}, \hat{x})$. Now if there is an A or W change below $\psi(x)$ then we get to lift $\gamma^{\hat{\rho}}(\hat{z})$ to be large at the next $\hat{\tau}$-expansionary stage. Since $\hat{\tau} \geq \tau\hat{\ }\infty$ then we have seen $\Psi(A, W, Q, x)$ recover; hence we can ensure $\gamma^{\hat{\rho}}(\hat{z}) > \psi(x)$. We can lift $\gamma^{\hat{\rho}}(\hat{z})$ to be large when this happens because $\hat{\tau}$ can see that $\Gamma(A, W)$ is totally ω^2-c.a.; hence we know this happens finitely often and so this cannot send the use to infinity.

A τ node guesses whether $\Psi(A, W, Q)$ is total by measuring the length of the domain. But notice that it could be that $\operatorname{dom}\Psi(A, W, Q)$ goes to infinity but there could be an input with unbounded use. If this happens then $\Psi(A, W, Q)$ is not total. In this situation it is clear that $\Gamma(A, W)$ will not be ω^2-c.a. Above we discussed that along the path containing the finite outcome of all the children of a P or R requirement shows that $A \oplus W$ is not totally ω^2-c.a. but this is not entirely true since it could be that $\Psi(A, W, Q)$ is actually not total due to the reason above. Therefore we see we also need nodes that check whether there is an input with unbounded use; this is a $\mathbf{0}'''$ feature of the construction. We cannot measure this using a single node (unless we allow nodes with $\omega+2$ outcomes so we spread nodes, denoted ζ_k, down the tree each measuring whether the computation $\Psi(A, W, Q, k)$ converges or not for a fixed k. We call these nodes *sons* and the children discussed previously are referred to as *daughters*. We place son nodes down with k increasing as we go down the tree; hence we are also able to determine the least point where $\Psi(A, W, Q)$ diverges. Notice that R requirements will also need to have sons nodes on the tree. Similarly they will measure whether the computation $\Theta(A, W, k)$ converges or not for a fixed k.

Now back to considering the interaction between the R and P requirements. We need to decrease $m(x)$ by one each time a number enters Q below $\psi(x)$. Then we need to know how many times this can happen when we established x. Let η be a node working for an R requirement and let τ be a node working for a P requirement. Let μ and ρ be children of η and τ respectively. When x is established it can look to see which $\mu > \rho\hat{\ }\infty$ currently have followers appointed and then this could tell us how many times we need to decrease $m(x)$ by one. For this to work we need to make sure that these are the only enumerations into Q that can injure $\Psi(A, W, Q, x)$. First consider the case where $\eta \succeq \rho\hat{\ }\infty$. Suppose μ is not allowed to injure $\Psi(A, W, Q, x)$, then μ needs to *protect* (ρ, x). Now $\eta \succeq \rho\hat{\ }\infty$ so η knows $\Gamma(A, W)$ is ω^2-c.a. Then we are able to define $\delta(p) \geqslant \gamma(z)$ because we know that this cannot drive the use to infinity. Now whenever $\Delta(A, W, p)\uparrow$ we are able to appoint a new large follower. So now whenever there is a change below $\gamma(z)$, we are able to appoint a new large follower; hence we can ensure $q > \psi(x)$. Now consider μ to the right of $\rho\hat{\ }\infty$. Now due to the way we arrange the tree, η must extend a child of τ. If η extends a daughter of τ, then η believes $\Gamma(A, W)$ is total so we are able to protect (ρ, x). But if η extends a son of τ, then η believes that $\operatorname{dom}\Psi(A, W, Q) = k$. In this case we are able to protect (ρ, x) if $x < k$. If $x \geqslant k$ then we are not able to protect because we cannot define $\delta(p) \geqslant \gamma(z)$ as this would drive $\delta(p)$ to infinity.

So consider the case where η believes that $\operatorname{dom}\Psi(A, W, Q) = k$ and $x \geqslant k$. If η is correct then the computation $\Psi(A, W, Q, x)$ changes infinitely often. We want to set things up so that the enumeration by μ does not corrupt $\Psi(A, W, Q, x)$. Now notice that μ is only accessible at a stage where there has been a change in this computation and μ is initialised at every ρ-expansionary stage (this is because μ is to the right of ρ). Then an enumeration by μ is *invisible* to ρ because this will only happen along with some other change since the last ρ-expansionary stage. So when it comes to the ordinal counting the Q change is invisible. But notice that $\Gamma(A, W, z)\downarrow$ at the stage μ enumerates q into Q. Then although the Q change did not cause a problem with the counting, the Q change still renders the current tracker useless; hence this enumeration still corrupts x. Now since the computation $\Psi(A, W, Q, x)$ changes infinitely often there are also infinitely many stages such that $\Gamma(A, W, z)[s]\uparrow$. So the strategy is to wait for a stage where $\Gamma(A, W, z)[s]\uparrow$ and then at this stage we can enumerate q into Q. This ensures that we do not need to replace the tracker because we are able to define $\Gamma(A, W, z)$ after the Q change. To do this μ sends a *request token* to τ at the stage μ sees $\Theta(A, W, q)[s] = 0$. Then at the next τ-expansionary stage such that there has been a change in the computation $\Psi(A, W, Q, x)$ since the last τ-expansionary stage then τ enumerates q into Q for μ and we leave $\Gamma(A, W, z)[s]\uparrow$. Notice that if μ is correct then we must eventually see a change in the computation $\Psi(A, W, Q, x)$, so we will eventually successfully enumerate q into Q. So when a new follower for μ is appointed after an enumeration into Q or after initialisation, we define a set $\mathtt{pro}(\mu)$ containing all (ρ, x) such that x has already been established and either μ extends some daughter of τ or μ extends a son of τ and $x < k$. Notice that when we have multiple P requirements τ may need to pass the request on to another $\hat{\tau}$ node if τ believes $\operatorname{dom}\Psi_{\hat{\tau}}(A, W, Q) = \hat{k}$.

When we first appoint the anchor for μ, p, we define the set of protected (ρ, x), $\mathtt{pro}_s(\mu)$. We redefine the set of protected (ρ, x) at any stage we appoint a new follower due to a failed diagonalisation attempt (this is a stage where we see an A or W change below $\delta(p)$ and the current follower is in Q). At stages where we redefine $\mathtt{pro}_s(\mu)$ we include all established (ρ, x) such that $\rho\hat{}\infty \preceq \mu$. Now we also halt and initialise weaker nodes when we do this. So notice that every time a protected set for any μ is defined, there is a different least element for every (ρ, x) such that $\rho\hat{}\infty \preceq \mu$. Now when μ enumerates a number into Q at stage s, for each ρ such that $\rho\hat{}\infty \preceq \mu$, we can declare the least x such that $(\rho, x) \notin \mathtt{pro}_s(\mu)$ to be corrupted and x can take over all established $x' > x$. Then because this least x is unique to this enumeration, every x is only corrupted once. We choose to declare the least unprotected x to be corrupted even when $q > \psi(x)$, to ensure x is only declared corrupted once. Also notice that if x is the least corrupted by an enumeration by μ at stage s then x was invented at the stage $\mathtt{pro}_s(\mu)$ was defined.

Consider two P subrequirements such that (ρ, x) is currently in an attack, $\hat{\rho} \succeq \rho\hat{}\infty$, $\hat{x}$ is corrupted and $(\hat{\rho}, \hat{x}) \notin \mathtt{pro}_s(\rho, x)$. Suppose (ρ, x) enumerates a number into A at stage s. $(\hat{\rho}, \hat{x}) \notin \mathtt{pro}_s(\rho, x)$ so $\hat{x}$ was established after $\mathtt{pro}_s(\rho, x)$ was defined. Note that nodes to the right of ρ are initialised at every stage $\gamma(z)$ is lifted large. Now $\hat{x}$ is corrupted at stage s so the μ that corrupts $\hat{x}$ must be accessible during the (ρ, x) attack; hence μ is to the right of ρ. This means that μ is initialised at every stage $\gamma(z)$ is lifted large. Since $\mathtt{pro}_t(\mu)$ was defined at the stage $\hat{x}$ was invented, it follows that $\gamma(z)$ was last lifted large before $\hat{x}$ was established. Notice that this means the number that (ρ, x) enumerates into A at stage s is relatively small to $(\hat{\rho}, \hat{x})$. So if we define the use of $\gamma^{\hat{\rho}}(\mathtt{tr}(\hat{x}))$ to be large at the first $\hat{\tau}$-expansionary stage after $\hat{x}$ was invented, then the enumeration at stage s uncorrupts $\hat{x}$. Now notice that by doing this if there is ever an A change below $\psi(x)$ while corrupted, this A change must be small enough to uncorrupt x.

z acts as a tracker for x when $\gamma_s(z) \geqslant \psi_s(x)$. So notice that z could act as a tracker for many x. We just need to ensure that $\Gamma_\Psi(A, W)$ is total, so we need to ensure z is the tracker for finitely many x. So now we consider z as a tracker for an *interval* rather than for a single input. Corruption and attacks are actions that can only happen finitely often for each x so we choose these stages to allow use to increase the size of the interval z acts as a tracker for. Each interval has a least element, we call all other inputs in the interval *taken over* by the least element.

7.3. **Tree of Strategies.** Consider a parent node (τ or η), below the finite outcome the parent does not have any children and below the infinite outcome we will start assigning nodes to be children of this parent node, alternating between daughters and sons. We call a parent node *closed* below the finite outcome of the parent as well as below the infinite outcome of a daughter and the divergent outcome of a son. Once the parent is *closed*, we stop assigning nodes to be children of this parent and we move to the next parent, alternating between τ and η. We need to ensure there is a node working for every N_Φ on every path, so to do this we assign every node of even length to work for an N requirement.

7.4. **Assigning Requirements.** List all functionals in order type ω. Then we assign requirements by induction. Let λ be the root of the tree. Assign λ to working for requirement P_Ψ where Ψ is the first functional on the list. Now let β be the longest node such that $\beta \prec \alpha$ and β is not a node working for N_Φ. Then we assign a requirement to α as follows:

Let α be a node of length l, then if l is odd then let γ be the longest node such that $\gamma \prec \alpha$ and γ is a node working for $N_{\hat{\Phi}}$. Now assign α to work for requirement N_Φ, where Φ is the next functional on the list after $\hat{\Phi}$. Otherwise find which of the following cases apply:

Case 1. β is a node working for P_Ψ.

- If $\beta\hat{}\infty \preceq \alpha$, then let α be a node working for requirement $P_{\Psi,0}$.
- If $\beta\hat{}\texttt{fin} \preceq \alpha$, then let α be a node working for requirement R_Θ where $\Theta = \Psi$.

Case 2. β is a node working for $P_{\Psi,k}$.

- If $\beta\hat{}\infty \preceq \alpha$, then let α be a node working for requirement R_Θ where $\Theta = \Psi$.
- If $\beta\hat{}\texttt{fin} \preceq \alpha$, then let α be a ζ_k node, son of P_Ψ.

Case 3. β is a ζ_k node, son of P_Ψ.

- If $\beta\hat{}\uparrow \preceq \alpha$, then let α be a node working for requirement R_Θ where $\Theta = \Psi$.
- If $\beta\hat{}\downarrow \preceq \alpha$, then let α be a node working for $P_{\Psi,k+1}$.

Case 4. β is a node working for R_Θ.

- If $\beta\hat{}\infty \preceq \alpha$, then let α be a node working for requirement $R_{\Psi,0}$.
- If $\beta\hat{}\texttt{fin} \preceq \alpha$, then let α be a node working for requirement P_Ψ where Ψ is the next functional on the list after Θ.

Case 5. β is a node working for $R_{\Theta,k}$.

- If $\beta\hat{}\infty \preceq \alpha$, then let α be a node working for requirement P_Ψ where Ψ is the next functional on the list after Θ.
- If $\beta\hat{}\texttt{fin} \preceq \alpha$, then let α be a ξ_k node, son of R_Θ.

Case 6. β is an ξ_k node, son of R_Θ.

- If $\beta\hat{}\uparrow \preceq \alpha$, then let α be a node working for requirement P_Ψ where Ψ is the next functional on the list after Θ.
- If $\beta\hat{}\downarrow \preceq \alpha$, then let α be a node working for $R_{\Theta,k+1}$.

7.5. **Types of Nodes.** There are seven types of nodes. These have been discussed in the technical discussion; here we give a summary of the action each node takes.

Nodes working for requirement N_Φ are denoted π and have outcomes $\infty < \texttt{fin}$. π nodes measure the length of the domain of $\Phi(A)$. At $\pi\hat{}\infty$ stages we *establish* a new input y. When y is established we define the first ordinal for the ω-computable approximation for $\Phi_\pi(A, y)$.

Nodes working for requirement R_Θ are denoted η. η nodes have outcomes $\infty < \texttt{fin}$. η nodes measure the length of the domain of $\Theta(A, W)$ and at stages where η believes $\Theta(A, W)$ is total it will define an initial segment of $\Delta_\eta(A, W)$. η also checks whether any of its children need their current follower replaced by a new one. Note that we need to do this at η because we need to define $\delta(p) \geqslant \theta(q)$, where p is the anchor and q is the follower.

Nodes working for requirement $R_{\Theta,k}$, denoted μ, are called *daughter* nodes, placed on the tree below its parent η. μ nodes have outcomes $\infty < \mathtt{fin}$. μ nodes have an anchor $p = \mathtt{ac}_s(\mu)$ and a follower $q = \mathtt{fl}_s(\mu)$ and enumerate numbers into Q to diagonalise. If μ extends the divergent outcome of a son of τ, μ will send a request for the enumeration of a follower into Q instead of doing the enumeration itself. When we appoint the first follower for μ we define the set of (ρ, x) that μ must *protect*, $\mathtt{pro}(\mu)$. Then at stages we appoint a new follower after a previous enumeration we redefine this set. When we enumerate a number into Q we declare (ρ, x) that are not protected by μ to be *corrupted*.

ξ_k nodes are called *sons* of η. A ξ_k node measures whether $\Theta_\eta(A, W, k)$ converges or diverges. ξ_k nodes have outcomes $\uparrow<\downarrow$.

Nodes working for requirement P_Ψ are denoted τ and have outcomes $\infty < \mathtt{fin}$. A τ node measures the length of the domain of $\Psi(A, W, Q)$. At stages where τ believes $\Psi(A, W, Q)$ is total it will define an initial segment of $\Gamma_\tau(A, W)$. τ also has the job of dealing with requests from μ nodes below the divergent outcome of a son of τ.

Nodes working for requirement $P_{\Psi,k}$, denoted ρ, are called *daughter* nodes, placed on the tree below the infinite outcome of its parent τ. ρ nodes have outcomes $\infty < \mathtt{fin}$. At ρ-expansionary stages a new input, x, is *invented*, this means we give x a tracker $\mathtt{tr}_s(x)$, this is the first tracker for x which we also denote $\mathtt{orig}(x)$. At the next ρ-expansionary stage x is *established*, at this stage we define the first ordinal for the ω^2-computable approximation for $\Psi(A, W, Q, x)$, $o^\Psi_t(x)$. So $z = \mathtt{tr}_s(x)$ means z is the tracker for x at stage s, but also the tracker for all inputs that have been taken over by x. When an input is taken over by x, it is added to the interval $I(x)$. So at stage s, z is the tracker for all inputs in $I_s(x)$. For action on the interval $I(x)$ we refer to action on the least input in that interval. $\mathcal{C}_s(\rho)$ denotes the collection of these inputs that have been established but not taken over. While x is corrupted (ρ, x) will start an attack if there is a W change below $\psi(I(x))$. While (ρ, x) is in an attack it will define the set of $(\hat{\rho}, \hat{x})$ that (ρ, x) must *protect*, $\mathtt{pro}(\rho, x)$; this will be updated after each enumeration into A. Now notice we only update the ordinal for x, $o^\Psi_s(x)$, at ρ-expansionary stages. This means that when x is declared corrupted, we do not decrease the ordinal until the next ρ-expansionary stage. So if there is a W change below $\psi_s(I(x))$ between the stage x is declared corrupted and the next ρ-expansionary stage we do not need to start an attack. At the first ρ-expansionary stage after x is declared corrupted we declare x to be *fully corrupted.* Note that we do not see the ordinal for the new tracker until x has been declared to be fully corrupted. Similarly, we do not see the new ordinal after finishing an attack until the next ρ-expansionary stage after the attack was declared finished. We declare a (ρ, x) attack to be finished at a τ node, so we will not see the new ordinal until the next ρ-expansionary stage; hence we do not want to start another attack between the stage the attack is declared finished and the next ρ-expansionary stage.

ζ_k nodes are called *sons* of τ. A ζ_k node measures whether $\Psi_\tau(A, W, Q, k)$ converges or diverges. ζ_k nodes have outcomes $\{\uparrow, \downarrow\}$ with $\uparrow<\downarrow$.

Note that we may assume that all computations we define have non-decreasing use.

7.6. **Construction.** At the beginning of each stage s we check if any (ρ, x) needs declaring uncorrupted or needs to start an attack. Let ρ be a node working for requirement $P_{\Psi,k}$ and let x be such that $x \in \mathcal{C}_s(\rho)$ and x is corrupted. If $\Gamma_\rho(A, W, \texttt{orig}(x))[s] \uparrow$, then define $\texttt{tr}_{s+1}(\rho, x) = \texttt{orig}(\rho, x)$ and declare x to be uncorrupted.

If (ρ, x) has not been uncorrupted then let $u = \psi^\rho_{s-1}(I_{s-1}(\rho, x))$. (ρ, x) wants to attack if all of the following hold:

- $W_{s-1} \upharpoonright u \neq W_s \upharpoonright u$
- (ρ, x) is not currently in an attack
- (ρ, x) is fully corrupted
- (ρ, x) was not in an attack at the last ρ-expansionary stage

If (ρ, x) wants to attack and there is no $(\hat{\rho}, \hat{x})$ to its left that wants to attack, then declare (ρ, x) to begin an attack.

If an attack has been started or an x has been declared uncorrupted then halt the stage and initialise all nodes to the right of $\rho\hat{\ }\infty$. For each such (ρ, x) declare all established $x' > x$ taken over by x. Formally, define $\mathcal{C}_{s+1}(\rho) = \{y \leqslant x : y \in \mathcal{C}_s(\rho)\}$ and for established $z \geqslant x$, $I_{s+1}(z) = \{y \in I_s(w) : w \in \mathcal{C}_s(\rho) \text{ and } w \leqslant x\}$, and for $z < x$ $I_{s+1}(z) = I_s(z)$.

If the stage has not been halted then let the collection of accessible nodes δ_s be an initial segment of the tree of strategies. δ_s is defined by recursion; the root of the tree is in δ_s, then the action of each node defines the next accessible node.

If a π node is accessible at stage s. Let $t < s$ be the last stage $\pi\hat{\ }\infty$ was accessible, $t = 0$ if there was no such stage. If $\operatorname{dom} \Phi_\pi(A_s) < \#t$ then $\pi\hat{\ }\texttt{fin}$ is accessible; otherwise $\pi\hat{\ }\infty$ is accessible. If $\pi\hat{\ }\infty$ is accessible, then declare the least y that has not already been established to be established.

If a τ node is accessible at stage s. Let $t < s$ be the last τ-expansionary stage, $t = 0$ if there was no such stage. If $\operatorname{dom} \Psi_\tau(A, W, Q)[s] < \#t$ then $\tau\hat{\ }\texttt{fin}$ is accessible; otherwise s is a τ-expansionary stage.

If s is a τ-expansionary stage and there is a request token on τ from a μ node, then $\mu \succ \zeta_k \uparrow$ for some son, ζ_k, of τ. If $\Delta_\mu(A, W, p)[s] \uparrow$ then cancel the request. If the request has not been cancelled and the computation $\Psi_\tau(A, W, Q, k)[t]$ does not hold at stage s, then:

- If $\texttt{prec}(\tau) = \varnothing$ then carry out the request by enumerating $q = \texttt{fl}_s(\mu)$ into Q_{s+1}. Declare (ρ, x) corrupted if $\rho\hat{\ }\infty \prec \mu$, $x \in \mathcal{C}_s(\rho)$ and $(\rho, x) \notin \texttt{pro}_s(\mu)$. For each ρ let x be the least declared corrupted, then declare all established $x' > x$ taken over by x. If (ρ, x) has been declared corrupted and has not been taken over, then define a new large tracker $\texttt{tr}_{s+1}(\rho, x)$ for x, halt and initialise all nodes weaker than $\mu\hat{\ }\infty$.
- Otherwise move the request token to the longest $\hat{\tau} \in \texttt{prec}(\tau)$, halt and initialise all nodes weaker than $\mu\hat{\ }\infty$.

If the stage has not been halted, then let ρ be a daughter of τ and $z = \texttt{tr}_s(\rho, x)$ for $x \in \mathcal{C}_s(\rho)$. If $\Gamma_\tau(A, W, z)[s] \uparrow$ then do the first of the following that applies:

- If (ρ, x) is not in an attack then:

  - If x was invented at stage t, define $\Gamma_{\tau,s+1}(A_s, W_s, z) = s$ with large use.
  - Otherwise define $\Gamma_{\tau,s+1}(A_s, W_s, z) = s$ with use $\psi^{\rho}_s(I_s(\rho, x))$.
- Let r be the stage (ρ, x) began its attack. If $d^{\rho}_r(z) > d^{\rho}_s(z)$ then define $\texttt{pro}_{s+1}(\rho, x)$ to be the empty set, declare the (ρ, x) attack to be finished and define $\Gamma_{\tau,s+1}(A_s, W_s, z) = s$ with use $\psi^{\rho}_s(I_s(\rho, x))$.
- If (ρ, x) was not in an attack at stage t (the last τ-expansionary stage), then declare (ρ, x) to be fully in an attack and define $\Gamma_{\tau,s+1}(A_s, W_s, z) = s$ with large use and define $\texttt{pro}_{s+1}(\rho, x)$ to be the set of $(\hat{\rho}, \hat{x})$ such that $\hat{x} \in \mathcal{C}_s(\hat{\rho})$ and $\hat{\rho}\hat{\ }\infty \prec \tau$, halt and initialise all nodes to the right of $\rho\hat{\ }\infty$.
- If (ρ, x) enumerated a number into A at stage t, then define $\texttt{pro}_{s+1}(\rho, x)$ to be the set of $(\hat{\rho}, \hat{x})$ such that $\hat{x} \in \mathcal{C}_s(\hat{\rho})$ and $\hat{\rho}\hat{\ }\infty \prec \tau$, and define $\Gamma_{\tau,s+1}(A_s, W_s, z) = s$ with large use, halt and initialise all nodes to the right of $\rho\hat{\ }\infty$
- if there is some $(\hat{\rho}, \hat{x}) \in \texttt{pro}_s(\rho, x)$ with tracker $\texttt{tr}_t(\hat{\rho}, \hat{x}) = \hat{z}$ such that the computation $\Gamma_{\hat{\rho}}(A, W, \hat{z}))[t]$ no longer holds at stage s, or $\hat{x}$ has been uncorrupted between stages t and s, then define $\Gamma_{\tau,s+1}(A_s, W_s, z) = s$ with large use, halt and initialise all nodes to the right of $\rho\hat{\ }\infty$.
- Otherwise define $\Gamma_{\tau,s+1}(A_s, W_s, z) = s$ with use $\gamma_{t+1}(z)$.

For $z < t$ if $\Gamma_\tau(A, W, z)[s] \uparrow$ and z is not a tracker for any daughter of τ, then define $\Gamma_{\tau,s+1}(A_s, W_s, z) = 0$ with use 0. If the stage has not been halted then let $\tau\hat{\ }\infty$ be accessible.

If a ρ node working for requirement $P_{\Psi,k}$ is accessible at stage s. Let $t < s$ be the last ρ-expansionary stage, $t = 0$ if there is no such stage. If $\forall z < \#t$, $\Gamma_\tau(A, W, z)[s] = f^{\rho}_s(z)$ and $o^{\rho}_s(z) < \omega^2$, then s is a ρ-expansionary stage; otherwise $\rho\hat{\ }\texttt{fin}$ is accessible.

Suppose s is a ρ-expansionary stage. Let $x \in \mathcal{C}_s(\rho)$ and let $z = \texttt{tr}_s(x)$. Then do the following:

- If (ρ, x) was either corrupted at stage t then declare x to be fully corrupted.
- If (ρ, x) is in an attack then enumerate $\gamma_s(z)$ into A_{s+1}, halt the stage and initialise all nodes to the right of $\rho\hat{\ }\infty$.

If the stage has not been halted then invent a new input by letting $x + 1$ be least such that $\texttt{tr}_s(x + 1) \uparrow$ then define a new large tracker for $x + 1$. Define $\texttt{orig}(\rho, x + 1) = \texttt{tr}_{s+1}(x + 1)$. If $x + 1 > 0$ then declare x established and define $\mathcal{C}_{s+1}(\rho) = \mathcal{C}_s(\rho) \cup \{x\}$ and $I_s(x) = \{x\}$. Let $\rho\hat{\ }\infty$ be accessible.

If a ζ_k node, son of τ, is accessible at stage s. Let $t < s$ be the last $\zeta_k \uparrow$ stage, $t = 0$ if there is no such stage. If the computation $\Psi_\tau(A, W, Q, k)[t]$ does not hold at stage s then let $\zeta_k \uparrow$ be accessible; otherwise let $\zeta_k \downarrow$ be accessible.

If an η node is accessible at stage s. Let $t < s$ be the last η-expansionary stage, $t = 0$ if there is no such stage. If $\operatorname{dom}\Theta(A, W)[s] < \#t$ then $\eta\hat{\ }\texttt{fin}$ is accessible; otherwise s is an η-expansionary stage.

Suppose s is an η-expansionary stage. If $\Delta_\eta(A, W, p)[s] \uparrow$ and $p < s$ is not the anchor of any daughter of η, then define $\Delta_{\eta,s+1}(A_s, W_s, p) = 0$ with use 0. Otherwise, if $\Delta_\eta(A, W, p)[s] \uparrow$ and $p = \texttt{ac}_s(\mu)$ for μ a daughter of η, then let $q = \texttt{fl}_s(\mu)$ and do the first of the following that applies:

- If $q \in Q_s$ then cancel the follower q, appoint a new large follower, and leave $\Delta_\eta(A, W, p) \uparrow$. Define $\mathtt{pro}_{s+1}(\mu)$ to be the set of (ρ, x) such that (ρ, x) needs protection from μ at stage s. Halt and initialise all nodes weaker than $\mu\hat{}\infty$.
- If $q < \psi^\rho_s(I_s(\rho, x))$ for some $(\rho, x) \in \mathtt{pro}_s(\mu)$ then cancel the follower q and appoint a new large follower and leave $\Delta_\eta(A, W, p) \uparrow$. Halt and initialise all nodes weaker than $\mu\hat{}\infty$.
- Otherwise define $\Delta_{\eta,s+1}(A_s, W_s, p) = s$ with use $\theta_s(q)$. Note that we do not halt the stage.

If the stage has not been halted then let $\eta\hat{}\infty$ be accessible.

If a μ node working for requirement $R_{\Theta,k}$ is accessible at stage s. Let $t < s$ be the last μ-expansionary stage, $t = 0$ if there is no such stage. If $\forall p < \#t$, $\Delta_\eta(A, W, p)[s] = f^\mu_s(p)$ and $o^\mu_s(p) < \omega^2$, then s is a μ-expansionary stage, otherwise $\mu\hat{}\mathtt{fin}$ is accessible. If s is a μ-expansionary stage then:

- If $\mathtt{ac}_s(\mu) \uparrow$ then define a new large anchor and a new large follower. Define $\mathtt{pro}_{s+1}(\mu)$ to be the set of (ρ, x) such that (ρ, x) needs protection from μ at stage s. Halt and initialise all nodes weaker than $\mu\hat{}\infty$.
- Let $q = \mathtt{fl}_s(\mu)$. If $q \notin Q_s$, $\Theta(A, W, q)[s] = 0$ and $\mathtt{prec}(\mu) = \varnothing$, then enumerate q into Q_{s+1}. Declare (ρ, x) corrupted if $\rho\hat{}\infty \prec \mu$, $x \in \mathcal{C}_s(\rho)$ and $(\rho, x) \notin \mathtt{pro}_s(\mu)$. For each ρ let x be the least declared corrupted, then declare all established $x' > x$ taken over by x. If (ρ, x) has been declared corrupted and has not been taken over, then define a new large tracker, halt and initialise all nodes weaker than $\mu\hat{}\infty$.
- If $q \notin Q_s$, $\Theta(A, W, q)[s] = 0$ and $\mathtt{prec}(\mu) \neq \varnothing$, then send a request token for μ to the longest $\tau \in \mathtt{prec}(\mu)$, halt and initialise all nodes weaker than $\mu\hat{}\infty$.

If the stage has not been halted then let $\mu\hat{}\infty$ be accessible.

If a ξ_k node, son of η, is accessible at stage s. Let $t < s$ be the last $\xi\hat{}\uparrow$ stage, $t = 0$ if there is no such stage. If the computation $\Theta_\eta(A, W, k)[t]$ does not hold at stage s, then let $\xi_k\hat{}\uparrow$ be accessible; otherwise let $\xi_k\hat{}\downarrow$ be accessible.

7.7. **Verification.** First note that if $(\rho, x) \in \mathtt{pro}_s(\alpha)$ then $\tau\hat{}\infty \preceq \alpha$, so $\psi^\rho_s(I_s(\rho, x)) \downarrow$; hence the uses defined in the construction exist.

Lemma 7.1. *Let μ be a node working for requirement $R_{\Theta,k}$ with parent η. For all stages s, if $q = \mathtt{fl}_s(\mu)$ is defined then $p = \mathtt{ac}_s(\mu)$ is also defined, and if $\Delta_\eta(A, W, p)[s] \downarrow$ then $\Theta_\eta(A, W, q)[s] \downarrow$ and $\theta_s(q) \leqslant \delta_s(p)$.*

Proof. The anchor only becomes undefined at initialisation. Note that at the first μ-expansionary stage after initialisation an anchor and a follower is appointed. Then at any stage at which a new follower is appointed either a new anchor is also appointed or the anchor is already defined.

Suppose $\Delta_\eta(A, W, p)[s] \downarrow$ and let $t < s$ be the stage at which this computation was defined, and u be the use of this computation. Then stage t is an η-expansionary stage, so $\Theta_\eta(A, W, q)[t] \downarrow$. The computation $\Delta_\eta(A, W, p)$ has not changed between stages t and s so $A_s \upharpoonright u = A_t \upharpoonright u$ and $W_s \upharpoonright u = W_t \upharpoonright u$. At stage t, u was defined so that $u \geqslant \theta_t(q)$; hence $A_s \upharpoonright \theta_t(q) = A_t \upharpoonright \theta_t(q)$ and $W_s \upharpoonright \theta_t(q) = W_t \upharpoonright \theta_t(q)$. Then it follows that the use of $\Theta_\eta(A, W, q)[t]$ has not changed, so $\Theta_\eta(A, W, q)[s]\downarrow$ and $\theta_s(q) \leqslant \delta_s(p)$. □

Lemma 7.2. Let τ be a node working for requirement P_Ψ. Let s be the stage a request is carried out and $t \leqslant s$ be the stage τ acts on the request, then there are no τ-expansionary stages between stage t and stage s.

Proof. If $\mathtt{prec}(\tau) = \varnothing$ then τ carries out the request; hence $s = t$ and we are done. So we suppose $\mathtt{prec}(\tau) \neq \varnothing$, then at stage t, τ passes the request to a node $\hat{\tau}$ which has a son, $\hat{\zeta}_{\hat{k}}$, with $\hat{\zeta}_{\hat{k}} \uparrow\preceq \tau$. Then τ is only accessible at $\hat{\zeta}_{\hat{k}} \uparrow$ stages, so there has been a change in the computation $\Psi_{\hat{\rho}}(A, W, Q, \hat{k})$ between two τ stages. Then while $\hat{\tau}$ has the request any such changes would cause the request to be acted on, and when this happens the stage is halted. Let t_0 be the stage $\hat{\tau}$ acts on the request, then there are no τ-expansionary stages between stage t and t_0. If $t_0 = s$ we are done; otherwise it is passed to some τ'. Let t_1 be the stage τ' acts on the request; then we repeat the same argument to find there are no $\hat{\tau}$ stages between stage t_0 and t_1. If $t_1 = s$ then since every τ stage is a $\hat{\tau}$ stage we are done. Continue this argument, since $\mathtt{prec}(\tau)$ is finite and the request is carried out at stage s, $t_n = s$ for some n; hence there are no τ-expansionary stages between stage t and stage s. □

Lemma 7.3. For all stages s, **(1)**, **(2)**, and **(3)** hold.

- **(1):** Let ρ be a node working for requirement $P_{\Psi,k}$ with parent τ. Let $z = \mathtt{tr}_s(\rho, x)$ for $x \in \mathcal{C}_s(\rho)$. Suppose (ρ, x) is not in an attack at stage s and $\Gamma_\tau(A, W, z)[s]\downarrow$. Let t be the stage the computation $\Gamma_\tau(A, W, z)[s]$ was defined. Then $Q_t \upharpoonright \psi_t(I_t(x)) = Q_s \upharpoonright \psi_t(I_t(x))$ and for all $\hat{x} \in I_s(\rho, x)$, $\Psi(A, W, Q, \hat{x})[s]\downarrow$ and $\psi_s(I_s(\rho, x)) \leqslant \gamma_s(z)$.
- **(2):** Let μ be a node working for requirement $R_{\Theta,k}$ and let $q = \mathtt{fl}_s(\mu)$. Suppose ρ is a node such that $\rho\hat{\ }\infty$ to the left of μ such that $x \in \mathcal{C}_s(\rho)$ and $(\rho, x) \notin \mathtt{pro}_s(\mu)$. If q is enumerated into Q_{s+1} at stage s and $q < \psi^\rho_s(I_s(\rho, x))$, then either (ρ, x) is in an attack at stage s or $\Gamma_\tau(A, W, \mathtt{tr}(x))[s]\uparrow$.
- **(3):** Let μ be a node working for requirement $R_{\Theta,k}$, and let $p = \mathtt{ac}_s(\mu)$ and $q = \mathtt{fl}_s(\mu)$. If $q \notin Q_s$ and there is a $(\rho, x) \in \mathtt{pro}_s(\mu)$ such that $q < \psi^\rho_s(I_s(\rho, x))$, then $\Delta_\eta(A, W, p)[s]\uparrow$ and q is not enumerated into Q_{s+1} at stage s.

Proof. We prove the lemma by simultaneous induction on the stage s. Clearly at $s = 0$, **(1)**, **(2)** and **(3)** hold. So let $s > 0$ and suppose **(1)**, **(2)** and **(3)** hold for all stages $t < s$.

Consider **(1)** at stage s. Suppose $Q_t \upharpoonright \psi_t(I_t(x)) \neq Q_s \upharpoonright \psi_t(I_t(x))$. Then at stage $r \in [t, s)$ some μ enumerated its follower q into Q_{r+1} and $q < \psi_t(x)$. Note that $\Gamma_\rho(A, W, \mathtt{tr}(x))[r]\uparrow$ at a stage r where a new tracker is appointed. Therefore $z = \mathtt{tr}_s(x) = \mathtt{tr}_t(x)$ and so x has not been corrupted between stages t and s. Then since $x \in \mathcal{C}_s(\rho)$ it is also the case that x has not been taken over between stages t and s. Recall that by the construction, if $\mu \succeq \rho\hat{\ }\infty$ and $(\rho, x) \notin \mathtt{pro}_r(\mu)$ then x is either corrupted or taken over; hence either $(\rho, x) \in \mathtt{pro}_r(\mu)$ or μ is to the right of $\rho\hat{\ }\infty$. Now note that an attack is only declared finished at a stage where $\Gamma_\rho(A, W, z)\uparrow$. Then it follows that since (ρ, x) is not in an attack at stage s, it is also the case that (ρ, x) is not in an attack at any stage r such that $r \in [t, s]$.

(3) holds at stage r; hence $(\rho, x) \notin \mathtt{pro}_r(\mu)$. **(2)** holds at stage r; hence either $\Gamma_\rho(A, W, \mathtt{tr}(x))[r] \uparrow$ or (ρ, x) is in an attack at stage r. But (ρ, x) is not in an attack at any stage r such that $r \in [t, s]$ and $\Gamma_\rho(A, W, \mathtt{tr}(x))[r] \downarrow$, a contradiction. Hence $Q_t \upharpoonright \psi_t(I_t(x)) = Q_s \upharpoonright \psi_t(I_t(x))$. Let u be the use of the computation $\Gamma_\tau(A, W, z)[s]$. At stage t, the use u was defined to be $\psi_t(I_t(x))$ so $(A, W, Q)_t \upharpoonright \psi_t(I_t(x)) = (A, W, Q)_s \upharpoonright \psi_t(I_t(x))$. Then it follows that for all $\hat{x} \in I_s(x)$, $\Psi(A, W, Q, \hat{x})[s] \downarrow$ and $\psi_s(I_s(x)) \leqslant \gamma_s(z)$. Thus **(1)** holds at stage s.

Consider **(2)** at stage s. First, if (ρ, x) is in an attack at stage s then we are done. So suppose (ρ, x) is not in an attack at stage s. Let τ be the parent of ρ and let r_0 be the stage $\mathtt{pro}_s(\mu)$ was defined. Notice that μ is initialised at every ρ-expansionary stage; hence x was established before stage r_0. $(\rho, x) \notin \mathtt{pro}_s(\mu)$; therefore either μ believes $\operatorname{dom} \Psi_\rho(A, W, Q) = k'$ and $I_{r_0}(x) \nsubseteq k'$, or μ lies to the right of $\tau\hat{\ }\infty$.

Case 1. μ lies to the right of $\tau\hat{\ }\infty$. Then μ is initialised at every τ-expansionary stage; hence we do not see any $\Psi_\tau(A, W, Q)$ computations recover while q is the follower of μ. Let r_1 be the stage q is appointed. At stage r_1, q is appointed to be large; therefore for all x' such that $\Psi_\tau(A, W, Q, x')[r_1] \downarrow$, $q > \psi_{r_1}(x')$. If there are any changes below the use of any of these computations while q is the follower for μ then they do not recover until after stage s. So it follows that for all x' such that $\Psi_\tau(A, W, Q, x')[s] \downarrow$, $q > \psi_s(x')$. So it is not possible to have $q < \psi^\rho_s(I_s(\rho, x))$ at stage s.

Case 2. μ believes $\operatorname{dom} \Psi_\rho(A, W, Q) = k'$ and $I_{r'}(x) \nsubseteq k'$. Now $\tau \in \mathtt{prec}(\mu)$ so $\mathtt{prec}(\mu) \neq \varnothing$. Therefore enumeration of q into Q_{s+1} at stage s is carried out by a request token at some $\hat{\tau}$ (note that it could be that $\hat{\tau} = \tau$). Let $t \leqslant s$ be the stage τ acts on the request and $r < t$ be the stage τ receives the request. Then stage t is the first τ-expansionary stage after a change in the computation $\Psi_\rho(A, W, Q, k')[r]$. Notice that if there is a stage $t > r'$ such that $I_{r'}(x) \neq I_t(x)$ then μ is initialised; hence $I_s(x) \nsubseteq k'$. Now let x' be the least element in the interval $I_t(k')$. Suppose (ρ, x') is in an attack at stage t. Note that μ is not initialised between stages r_0 and s, so this attack started before stage r_0. Then at the last ρ-expansionary stage a number is enumerated into A and all established inputs greater than x' are taken over by x'. Then $x = x'$, but an attack can only be declared finished at τ-expansionary stages, and by Lemma 7.2 there are no τ-expansionary stages between stage t and s; hence if (ρ, x') is in an attack at stage t then (ρ, x) is in an attack at stage s, a contradiction. Then (ρ, x') is not in an attack at stage t, and **(1)** holds at stage t, so it follows that $\psi_t(k') \leqslant \gamma_{t+1}(\mathtt{tr}_{t+1}(x')) \leqslant \gamma_{t+1}(\mathtt{tr}_{t+1}(x))$. Then the change in the computation $\Psi_\rho(A, W, Q, k')[r]$ between stages r and t causes $\Gamma_\rho(A, W, \mathtt{tr}(x))[t] \uparrow$. t is the first τ-expansionary stage after stage r; at this stage the request is acted on and we do not define $\Gamma_\rho(A, W, \mathtt{tr}(x))[t]$. $\Gamma_\rho(A, W, \mathtt{tr}(x))[s]$ is only defined at τ-expansionary stages, and by Lemma 7.2 there are no such stages between stage t and stage s; hence $\Gamma_\rho(A, W, \mathtt{tr}(x))[s] \uparrow$.

Consider **(3)** at stage s. Suppose at stage s there is a node μ with follower q and anchor p, and there is a $(\rho, x) \in \mathtt{pro}_s(\mu)$ such that $q < \psi^\rho_s(I_s(\rho, x))$. If $\Delta_\mu(A, W, p)[s] \uparrow$ and $\mathtt{prec}(\mu) \neq \varnothing$ then the enumeration is carried out by a τ node acting on a request, but since $\Delta_\mu(A, W, p)[s] \uparrow$ the request will be cancelled; hence q is not enumerated into Q_{s+1}. If $\Delta_\mu(A, W, p)[s] \uparrow$ and $\mathtt{prec}(\mu) = \varnothing$ then q is enumerated into Q_{s+1} at an η-expansionary stage, but $\Delta_\mu(A, W, p)[s] \uparrow$ and there is a $(\rho, x) \in \mathtt{pro}_s(\mu)$ such that $q < \psi_s^\rho(I_s(\rho, x))$, so q will be cancelled before it can be enumerated into Q_{s+1}. Therefore it suffices to show that $\Delta_\mu(A, W, p)[s] \uparrow$.

So suppose $\Delta_\mu(A, W, p)[s] \downarrow$ and let t be the stage this computation was defined. Let r_0 be the stage $\mathtt{pro}_s(\mu)$ was defined, and let r_1 be the stage q was appointed (note that $r_0 \leqslant r_1$). x was established before stage r_1 and $\tau\hat{\ }\infty \preceq \eta$, so it follows that $q > \psi_{r_1}^\rho(I_{r_1}(x))$. Notice that at stage t, $\Delta_\mu(A, W, p)[t] \uparrow$. If $q < \psi_t^\rho(I_t(x))$ we would not define $\Delta_\mu(A, W, p)[t+1]$, instead we would cancel q, but $q = \mathtt{fl}_s(\mu)$; hence $q > \psi_t^\rho(I_t(x))$. Then at stage t we define $\Delta_\mu(A, W, p)[s]$ with use $\theta(q)$, and since $\theta(q) \geqslant q$ it follows that $\delta_s(p) > \psi_t^\rho(I_t(x))$. Now at stage s, $q < \psi_s^\rho(I_s(x))$ so $Q_t \upharpoonright \psi_t^\rho(I_t(x)) \neq Q_s \upharpoonright \psi_t^\rho(I_t(x))$. Let $\hat{\mu}$ be the node that enumerates its follower $\hat{q}$ into Q_{r+1} at stage $r \in [t, s)$. Notice $q \notin Q_s$ and $q = \mathtt{fl}_s(\mu) = \mathtt{fl}_t(\mu)$, so $\hat{\mu} \neq \mu$.

Now μ has not been initialised between stages t and s; hence $\hat{\mu}$ is weaker than μ. Let r_2 be the stage that $\mathtt{pro}_r(\hat{\mu})$ was defined. Notice $r_2 > r_1$ because $\hat{\mu}$ is initialised when q is appointed. Now this means x was already established at stage r_2. If $(\rho, x) \in \mathtt{pro}_r(\hat{\mu})$, $r < s$ so **(3)** holds at stage r; hence $\hat{q}$ is not enumerated into Q_{r+1}. But $\hat{q}$ is enumerated into Q_{r+1}; therefore $(\rho, x) \notin \mathtt{pro}_r(\hat{\mu})$. But x was established before stage r_2, so $\hat{\mu}$ lies to the right of $\rho\hat{\ }\infty$, $\hat{\mu}$ believes $\operatorname{dom} \Psi_\rho(A, W, Q) = k'$ and $I_{r_2}(\rho, x) \nsubseteq k'$. Now **(2)** holds at stage r, so either (ρ, x) is in an attack at stage r or $\Gamma_\rho(A, W, z)[r] \uparrow$.

Suppose (ρ, x) is in an attack at stage r. Since $(\rho, x) \in \mathtt{pro}_s(\mu)$, (ρ, x) was not in an attack at stage r_0; hence this (ρ, x) attack starts at stage $r_3 \in (r_0, r)$. Notice that this attack is prompted by a W change below $\psi(I(x))$, as discussed above $\delta_s(p) > \psi_t^\rho(I_t(x))$ and the computation $\Delta_\mu(A, W, p)[s]$ was defined at stage t; hence this attack was started before stage t. Now if μ was to the right of $\rho\hat{\ }\infty$ then it would be initialised at stage r_3 but $r_3 \in (r_0, r)$ so $\mu \succ \rho\hat{\ }\infty$ and hence $\eta \succeq \rho\hat{\ }\infty$. Then there are no η-expansionary stage until the attack is finished. (ρ, x) is still in an attack at stage r; hence $t > r$, but this is a contradiction because $r \in [t, s)$. Therefore $Q_t \upharpoonright \psi_t^\rho(I_t(x)) = Q_s \upharpoonright \psi_t^\rho(I_t(x))$ and so $q > \psi_s^\rho(I_s(x))$.

Therefore if $q < \psi_s^\rho(I_s(x))$ then $\Delta_\mu(A, W, p)[s] \uparrow$ and q is not enumerated into Q_{s+1} at stage s. □

Lemma 7.4. Let ρ be a node working for requirement $P_{\Psi,k}$ and let $x \in \mathcal{C}_s(\rho)$. Let μ be a node working for requirement $R_{\Theta,k}$, and let $q = \mathtt{fl}_s(\mu)$. Suppose q is enumerated into Q_{s+1} at stage s. If (ρ, x) is not in an attack, $q < \psi_s(I_s(\rho, x))$ and $\Gamma_\tau(A, W, \mathtt{tr}(x))[s] \downarrow$, then $\mu \succ \rho\hat{\ }\infty$ and $(\rho, x) \notin \mathtt{pro}_s(\mu)$.

Proof. If $(\rho, x) \in \mathtt{pro}_s(\mu)$ then by Lemma 7.3 **(3)**, q is not enumerated into Q_{s+1}, a contradiction.

If $(\rho, x) \notin \mathtt{pro}_s(\mu)$ and μ is to the right of $\rho\hat{\ }\infty$, then μ is initialised at every ρ-expansionary stage; hence x was established before q was appointed. But $\Gamma_\tau(A, W, \mathtt{tr}(x))[s] \downarrow$ and (ρ, x) is not in an attack; this is a contradiction to Lemma 7.3 **(2)**. □

Lemma 7.5. Let ρ be a node working for requirement $P_{\Psi,k}$ and let $x \in \mathcal{C}_s(\rho)$. Suppose x is declared corrupted at stage s. If at stage $t > s$, (ρ, x) is not in an attack, and some μ enumerates a follower q into Q_{t+1} such that $q < \psi_t(I_t(\rho, x))$ and $\Gamma_\tau(A, W, \mathtt{tr}(x))[t] \downarrow$, then x is taken over at stage t.

Proof. (ρ, x) was declared corrupted at stage s. Then at stage s some $\hat{\mu}$ enumerated its follower $\hat{q}$ into Q_{s+1}, such that $\hat{\mu} \succ \rho\hat{\ }\infty$ and x was the least input for ρ such that $(\rho, x) \notin \mathtt{pro}_s(\hat{\mu})$. At stage $t > s$, μ enumerates q into Q_{t+1} such that $q < \psi_t(I_t(\rho, x))$

and $\Gamma_\tau(A, W, \mathtt{tr}(x))[t]\downarrow$. It follows from Lemma 7.4 that $\mu > \rho^\frown\infty$. The stage is halted when a protected set is defined; hence $\mathtt{pro}_s(\hat\mu)$ and $\mathtt{pro}_t(\mu)$ were defined at different stages; call these stages $\hat r$ and r respectively. x was the least input that was not yet established at stage $\hat r$. If $r > \hat r$ then since $\mu > \rho^\frown\infty$, a new input has been established since stage $\hat r$; hence x has already been established at stage when we define $\mathtt{pro}_t(\mu)$ at stage r; hence $(\rho, x) \in \mathtt{pro}_t(\mu)$, but this contradicts Lemma 7.3 **(3)**. Now suppose $r < \hat r$. Let x' be the least input for ρ that was not yet established at stage r. $\hat\mu > \rho^\frown\infty$, so $\hat r$ is a ρ-expansionary stage; hence at stage $\hat r$ a new input is established. So $x' \neq x$. Then x is not the least input that is not in $\mathtt{pro}_s(\mu)$; hence at stage t, x is taken over. □

Lemma 7.6. Let ρ and $\hat\rho$ be nodes working for requirements $P_{\Psi,k}$ and $P_{\hat\Psi,\hat k}$ respectively. Let $x \in \mathcal{C}_s(\rho)$ and $\hat x \in \mathcal{C}_s(\hat\rho)$, with trackers z and $\hat z$ respectively. Let $(\hat\rho, \hat x)$ be in an attack at stage s, and let t be the stage the $(\hat\rho, \hat x)$ attack is finished. If $(\rho, x) \in \mathtt{pro}_s(\hat\rho, \hat x)$ then x is not corrupted at stage $r \in [s, t)$.

Proof. $(\rho, x) \in \mathtt{pro}_s(\hat\rho, \hat x)$ so $\rho^\frown\infty \prec \hat\rho$. x is corrupted by the enumeration of a follower of μ, q, into Q_{r+1} at stage r such that $\mu > \rho^\frown\infty$. Then consider the following cases:

Case 1. $\mu > \hat\rho^\frown\infty$. Then μ is not accessible during the $(\hat\rho, \hat x)$ attack; hence x is not corrupted at stage $r \in [s, t)$.

Case 2. $\mu^\frown\infty \prec \hat\rho$ or μ is to the left of $\hat\rho$. Then when x is corrupted at stage r, $\hat\rho$ is initialised causing the attack to stop; hence $t = r$.

Case 3. μ is to the right of $\hat\tau$. $(\rho, x) \in \mathtt{pro}_s(\hat\rho, \hat x)$, so x was already established at the stage $\mathtt{pro}_s(\hat\rho, \hat x)$ was defined and at this stage μ is initialised. Now if x was corrupted after stage s then the corrupting q was appointed after stage s, but $x \in \mathcal{C}_s(\rho)$ so $(\rho, x) \in \mathtt{pro}_r(\mu)$ and so x is not corrupted after stage s.

Case 4. $\mu > \hat\tau^\frown\infty$ and μ is to the right of ρ. Then q was appointed at a $\hat\tau$-expansionary stage. Consider the stage $\mathtt{pro}_s(\hat\rho, \hat x)$ was defined. Then at the last $\hat\tau$-expansionary stage before we define $\mathtt{pro}_s(\hat\rho, \hat x)$ either $(\hat\rho, \hat x)$ was not in an attack or $(\hat\rho, \hat x)$ enumerated a number into A. Then μ was initialised at this stage. $\mu > \hat\tau^\frown\infty$ so $\mathtt{pro}_r(\mu)$ was defined after (or at) the stage $\mathtt{pro}_s(\hat\rho, \hat x)$ was defined, but x was already established at this stage; hence $(\rho, x) \in \mathtt{pro}_r(\mu)$ and so x is not corrupted after stage s.

Therefore x is not corrupted at stage $r \in [s, t)$. □

Lemma 7.7. Let ρ be a node working for requirement $P_{\Psi,k}$ with parent τ. Let $z = \mathtt{tr}_s(\rho, x)$ for $x \in \mathcal{C}_s(\rho)$. For all stages s such that (ρ, x) is in an attack, if $\Gamma_\tau(A, W, z)[s]\downarrow$ then for all $(\hat\rho, \hat x) \in \mathtt{pro}_s(\rho, x)$, $\Gamma_{\hat\rho}(A, W, \mathtt{tr}(\hat\rho, \hat x))[s]\downarrow$ and $\gamma_s(z) \geqslant \gamma_s^{\hat\rho}(\mathtt{tr}_s(\hat\rho, \hat x))$.

Proof. Let s be the least counterexample. At stage s, $\Gamma_\tau(A, W, z)[s]\downarrow$ and there is some $(\hat\rho, \hat x) \in \mathtt{pro}_s(\rho, x)$ with tracker $\hat z$, such that $\Gamma_{\hat\rho}(A, W, \hat z)[s]\uparrow$ or $\gamma_s(z) < \gamma_s^{\hat\rho}(\hat z)$. Let t be the stage at which the computation $\Gamma_\tau(A, W, z)[s]$ was defined, and let u be the use of this computation. $\mathtt{pro}_s(\rho, x)$ is only redefined when $\Gamma_\tau(A, W, z)$ is defined, so $\mathtt{pro}_s(\rho, x) = \mathtt{pro}_t(\rho, x)$. $(\hat\rho, \hat x) \in \mathtt{pro}_s(\rho, x)$, so $\hat\tau^\frown\infty \prec \tau$; hence t is a $\hat\tau$-expansionary stage. Then at stage t, $\Gamma_{\hat\rho}(A, W, \mathtt{tr}_t(\hat x))[t]\downarrow$.

If u was defined to be large at stage t, then $\gamma_t(z) \geqslant \gamma_t^{\hat\rho}(\mathtt{tr}_t(\hat x))$. If u was not defined to be large, then it was defined to be $\gamma_r(z)$ where $r < t$ is the last τ-expansionary stage. s is the least counterexample, so at stage r $\gamma_r(z) \geqslant \gamma_r^{\hat\rho}(\mathtt{tr}_r(\hat x))$.

Note that by Lemma 7.6, $\hat{x}$ has not been corrupted between stages r and t. Since u was not defined to be large, it is also the case that $\hat{x}$ has not been uncorrupted between stages r and t; hence $\mathtt{tr}_r(\hat{x}) = \mathtt{tr}_t(\hat{x})$. If $\gamma_r^{\hat{\rho}}(\mathtt{tr}_t(\hat{x})) \neq \gamma_t^{\hat{\rho}}(\mathtt{tr}_t(\hat{x}))$ then the computation $\Gamma_{\hat{\rho}}(A, W, \mathtt{tr}(\hat{\rho}, \hat{x}))[r]$ no longer holds at stage t; hence we would define the use to be large, a contradiction. So at stage r, $\gamma_r^{\hat{\rho}}(\mathtt{tr}_t(\hat{x})) = \gamma_t^{\hat{\rho}}(\mathtt{tr}_t(\hat{x}))$, then it follows that $\gamma_t(z) \geqslant \gamma_t^{\hat{\rho}}(\mathtt{tr}_t(\hat{\rho}, \hat{x}))$.

Now the computation $\Gamma_\tau(A, W, z)[s]$ was defined at stage t, so $A_t \upharpoonright u = A_s \upharpoonright u$ and $W_t \upharpoonright u = W_s \upharpoonright u$. $u = \gamma_t(z) \geqslant \gamma_t^{\hat{\rho}}(\mathtt{tr}_t(\hat{x}))$, so it follows that the computation $\Gamma_{\hat{\rho}}(A, W, \mathtt{tr}_t(\hat{\rho}, \hat{x}))[t]$ has not changed between stage t and stage s. By Lemma 7.6, $\hat{x}$ has not been corrupted between stages t and s. $u \geqslant \gamma_t^{\hat{\rho}}(\mathtt{tr}_t(\hat{x})) > \gamma_t^{\hat{\rho}}(\mathtt{orig}(\hat{x}))$ therefore $\hat{x}$ has not been uncorrupted between stages t and s; hence $\mathtt{tr}_t(\hat{\rho}, \hat{x}) = \mathtt{tr}_s(\hat{\rho}, \hat{x}) = \hat{z}$. Then $\Gamma_{\hat{\rho}}(A, W, \hat{z})[s] \downarrow$ and $\gamma_s(z) \geqslant \gamma_s^{\hat{\rho}}(\hat{z})$. □

Lemma 7.8. Let μ be a node working for requirement $R_{\Theta,k}$. Let s and $t > s$ be successive μ-expansionary stages with $p = \mathtt{ac}_s(\mu) = \mathtt{ac}_t(\mu)$. Let $u = \delta_s(p)$. If $A_t \upharpoonright u \neq A_s \upharpoonright u$ or $W_t \upharpoonright u \neq W_s \upharpoonright u$, then $o_t^\mu(p) < o_s^\mu(p)$.

Proof. At stage $r \in (s, t)$, $\Delta_\eta(A, W, p)[r] \uparrow$, then at the next η-expansionary stage $r' \in (r, t]$, $\Delta_\eta(A, W, p)[r'] = r'$.

$$\Delta_\eta(A, W, p)[s] \leqslant s < r' = \Delta_\eta(A, W, p)[r'] \leqslant \Delta_\eta(A, W, p)[t]$$

so $\Delta_\eta(A, W, p)[s] \neq \Delta_\eta(A, W, p)[t]$. s and t are μ-expansionary stages, so $\Delta_\eta(A, W, p)[s] = f_s^\mu(p)$ and $\Delta_\eta(A, W, p)[t] = f_t^\mu(p)$, then $f_s^\mu(p) \neq f_t^\mu(p)$; hence $o_t^\mu(p) < o_s^\mu(p)$ since $\langle f_s^\mu, o_s^\mu \rangle$ is an ω^2-computable approximation. □

Lemma 7.9. Let μ be a node working for requirement $R_{\Theta,k}$. μ sends finitely many requests and enumerates finitely many followers into Q while a particular p is the anchor.

Proof. Suppose there are finitely many μ-expansionary stages. A request is sent at a μ expansionary stage; hence only finitely many requests are sent. If $\mathtt{prec}(\mu) = \varnothing$ then a follower is enumerated into Q at a μ-expansionary stage; hence only finitely many followers are enumerated into Q. If $\mathtt{prec}(\mu) \neq \varnothing$ then a follower is enumerated into Q when a request is carried out. There are only finitely many requests sent and only one follower can be enumerated into Q per request; thus finitely many followers are enumerated into Q.

Now suppose there are infinitely many μ-expansionary stages. Now if p is eventually cancelled, clearly only finitely many request are sent and finitely many followers are enumerated into Q while p is the anchor. So suppose p is never cancelled. Then μ is not initialised while p is the anchor. Once a follower has been enumerated into Q, a new follower is appointed only if there is a stage t where $\Delta_\mu(A, W, p)[t] \uparrow$. Note that when a new follower is appointed, the stage is halted. A follower is enumerated at a μ-expansionary; this stage is also an η-expansionary stage, so $\Delta_\mu(A, W, p)[s] \downarrow$ at the stage a follower is enumerated into Q. But infinitely many followers are enumerated into Q, so after each stage s where a follower is enumerated into Q, there is a stage t where $\Delta_\mu(A, W, p)[t] \uparrow$. Let r be the first μ-expansionary stage after stage t and let $u = \delta_s(p)$. Then $A_t \upharpoonright u \neq A_r \upharpoonright u$ or $W_t \upharpoonright u \neq W_r \upharpoonright u$, and by Lemma 7.8, $o_t^\mu(p) < o_s^\mu(p)$. There are infinitely many of these stages; then it follows that the ordinal $o^\mu(p)$ decreases infinitely often, but $\langle f_s^\mu, o_s^\mu \rangle$ is an ω^2-computable approximation, so this is a contradiction. Therefore finitely many numbers are enumerated into Q. Suppose μ sends infinitely many requests. Then only finitely many of these requests are carried out by the previous argument. μ is not initialised while p is the anchor, so these requests are cancelled due to $\Delta_\mu(A, W, p)\uparrow$. Then we can follow the same argument as above to show that this cannot happen as $\langle f^\mu_s, o^\mu_s\rangle$ is an ω^2-computable approximation. □

Lemma 7.10. Let μ be a node working for requirement $R_{\Theta,k}$. Then while a particular p is the anchor for μ, there are finitely many stages such that $\mathtt{pro}_s(\mu) \neq \mathtt{pro}_{s+1}(\mu)$.

Proof. If p is eventually cancelled, then clearly $\mathtt{pro}(\mu)$ can only be redefined finitely many times while p is the anchor. So let p be the anchor of μ that is never cancelled. While p is the anchor for μ, $\mathtt{pro}(\mu)$ gets redefined at a stage s where $\Delta_\eta(A, W, p)[s]\uparrow$ and $\mathtt{fl}_s(\mu) \in Q_s$. Notice that this means that a new follower is appointed at every stage that $\mathtt{pro}(\mu)$ gets redefined while p is the anchor. Then by Lemma 7.9, finitely many followers are enumerated into Q while p is the anchor; hence $\Delta_\eta(A, W, p)[s]\uparrow$ and $\mathtt{fl}_s(\mu) \in Q_s$, finitely often. Therefore we redefine $\mathtt{pro}(\mu)$ at finitely many stages. □

Lemma 7.11. Let ρ be a node working for requirement $P_{\Psi,k}$ with parent τ, such that $\rho\hat{\ }\infty$ is initialised finitely often. Then for all x there exists a stage t such that for all $s > t$, $I_s(\rho, x) = I_t(\rho, x)$.

Proof. First note that the interval $I(\rho, x)$ can only be redefined to be larger than it previously was. Suppose there are finitely many ρ-expansionary stages. Notice that finitely many inputs are ever established. Then there are only finitely many x' that could be added to the interval $I(\rho, x)$; hence it cannot change infinitely often.

Now suppose there are infinitely many ρ-expansionary stages. Then $\Gamma_\tau(A, W)$ is ω^2-c.a. Suppose there is some x such that the interval $I(\rho, x)$ changes infinitely often. ρ is initialised finitely often, so let r be the last stage that ρ is initialised. Then after stage r, x will change trackers at most three times (one when x is established, one when it is corrupted, and then back to the original tracker if x is ever uncorrupted). So let z be the last tracker for x. Now the interval $I(\rho, x)$ increases infinitely often, so the use $\psi_s(I_s(\rho, x))$ will go to infinity. By Lemma 7.3 **(1)**, for all stages s such that $z = \mathtt{tr}_s(x)$, $\gamma_s(z) \geqslant \psi_s(I_s(\rho, x))$. z is the last tracker of x (and is never cancelled), so the use $\gamma_s(z)$ must go to infinity. But this contradicts that $\Gamma_\tau(A, W)$ is ω^2-c.a. Therefore there cannot be an x such that its interval $I(\rho, x)$ changes infinitely often. □

Lemma 7.12. Let μ be a node working for requirement $R_{\Theta,k}$ with parent η. At all stages s at which $p = \mathtt{ac}_s(\mu)$, there are finitely many followers appointed.

Proof. If p is eventually cancelled then certainly only finitely many followers are appointed while p is the anchor. So let p be an anchor that is never cancelled. While a particular p is the anchor, a new follower is appointed at an η-expansionary stage s, where $\Delta_\eta(A, W, p)[s]\uparrow$ and either $\mathtt{fl}_s(\mu) \in Q_s$ or $q < \psi^\rho_s(I_s(\rho, x))$ for some $(\rho, x) \in \mathtt{pro}_s(\mu)$. Notice that if there are finitely many η-expansionary stages, then clearly only finitely many followers are appointed. So suppose there are infinitely many η-expansionary stages. At each η-expansionary stage such that $\Delta_\eta(A, W, p)[s]\uparrow$ and $\mathtt{fl}_s(\mu) \in Q_s$, we appoint a new follower. Then it follows that if this happens infinitely often, μ could enumerate infinitely many followers into Q

while p is the anchor, a contradiction to Lemma 7.9. By Lemma 7.10, since p is never cancelled, $\mathtt{pro}_s(\mu)$ eventually stabilises. Let $\mathtt{pro}(\mu)$ be the last protected set defined for μ while p is the anchor. Note $(\rho, x) \in \mathtt{pro}(\mu)$, $\rho\hat{}\infty$ is either to the left of μ or $\rho\hat{}\infty \prec \mu$. μ is initialised finitely often since p is never cancelled, so it follows that ρ is also initialised finitely often. Then by Lemma 7.11, for each $(\rho, x) \in \mathtt{pro}(\mu)$ the interval $I_s(\rho, x)$ eventually stabilises. Let $I(x)$ be the last interval defined for (ρ, x) while p is the anchor. Now if $(\rho, x) \in \mathtt{pro}(\mu)$ then either μ believes $\Psi_\tau(A, W, Q)$ is total or μ believes $\operatorname{dom}\Psi_\rho(A, W, Q) = k'$ for some k', and $I_r(\rho, x) \subseteq k'$ where r is the stage $\mathtt{pro}(\mu)$ was defined. We consider each case separately.

Case 1. $(\rho, x) \in \mathtt{pro}(\mu)$ and μ believes $\Psi_\tau(A, W, Q)$ is total, where τ is the parent of ρ. Then there is a daughter of τ, $\hat{\rho}$, such that $\hat{\rho}\hat{}\infty \preceq \mu$. Note that it could be that $\rho = \hat{\rho}$. There are infinitely many η-expansionary stages, so $\hat{\rho}\hat{}\infty$ is accessible infinitely often; hence $\Psi_\tau(A, W, Q)$ is total. Therefore the use $\psi^\rho(I(x))$ eventually stabilises; hence for $(\rho, x) \in \mathtt{pro}(\mu)$ such that μ believes $\Psi_\tau(A, W, Q)$ is total, we see $q < \psi_s(I_s(\rho, x))$ finitely often.

Case 2. $(\rho, x) \in \mathtt{pro}(\mu)$, μ believes $\operatorname{dom}\Psi_\rho(A, W, Q) = k'$ for some k', and $I_r(\rho, x) \subseteq k'$ where r is the stage $\mathtt{pro}(\mu)$ was defined. Notice that if there is a stage $t > r$ such that $I_r(x) \neq I_t(x)$, then μ would be initialised. But the assumption is that p is not cancelled after stage r; hence $I_r(x) = I(x)$. Now suppose we see $\mathtt{fl}_s(\mu) < \psi^\rho_s(I_s(\rho, x))$ at infinitely many η-expansionary stages. Let x' be the least input in the interval $I(x)$ such that the use $\psi_s(x')$ increases infinitely often. $I_r(x) \subseteq k'$ so $x' < k$. Then by the assignment of requirements, there is a brother of ρ, $\zeta_{x'}$ such that $\zeta_{x'} \downarrow\preceq \mu$. If the use $\psi(I(x))$ increases infinitely often then there must be infinitely many $\zeta_{x'}$ stages where the computation $\Psi_\rho(A, W, Q, x')$ has changed; hence there are infinitely many $\zeta_{x'} \uparrow$ stages. μ is initialised at every $\zeta_{x'} \uparrow$ stage, but μ is only initialised finitely often, a contradiction.

Therefore only finitely many followers are appointed while p is the anchor of μ. □

Lemma 7.13. Suppose η is accessible infinitely often and initialised finitely often, then either $\eta\hat{}\infty$ or $\eta\hat{}\mathtt{fin}$ is accessible infinitely often and initialised finitely often.

Proof. Suppose there are finitely many η-expansionary stages, then since η can only halt the stage at η-expansionary stages, $\eta\hat{}\mathtt{fin}$ is accessible infinitely often. Nodes extending $\eta\hat{}\infty$ are accessible finitely often and η is initialised finitely often, so $\eta\hat{}\mathtt{fin}$ is initialised finitely often.

Now suppose there are infinitely many η-expansionary stages, but finitely many $\eta\hat{}\infty$ stages, so let s be the stage such that for all $t > s$, t is not an $\eta\hat{}\infty$ stage. Anchors are only appointed at μ-expansionary stages, so after stage s no daughter of η is accessible. Then after stage s there are no new anchors appointed for any daughter of η. η must halt the stage infinitely often (in particular infinitely often after stage s). When η halts the stage, a new follower for some daughter of μ is appointed. Then since no new anchors are appointed and there are finitely many daughters of η with anchors, there is some μ such that a new follower is appointed infinitely often. But this is a contradiction to Lemma 7.12. Hence η halts the stage finitely often, so there are infinitely many $\eta\hat{}\infty$ stages. Since η is initialised finitely often and halts the stage finitely often, $\eta\hat{}\infty$ is initialised finitely often. □

Lemma 7.14. Suppose μ is accessible infinitely often and initialised finitely often, then either $\mu\hat{}\infty$ or $\mu\hat{}\mathtt{fin}$ is accessible infinitely often and initialised finitely often.

Proof. Suppose there are finitely many μ-expansionary stages. Then since μ can only halt the stage at a μ-expansionary stage, $\mu\hat{}\texttt{fin}$ is accessible infinitely often. Since nodes extending $\mu\hat{}\infty$ are accessible finitely often and μ is initialised finitely often, $\mu\hat{}\texttt{fin}$ is initialised finitely often.

Now suppose there are infinitely many μ-expansionary stages. μ is initialised finitely often, so there is an anchor that is never cancelled; let p be this anchor. μ halts the stage if μ enumerates its follower into Q or sends a request token. By Lemma 7.9, this happens finitely often; hence μ halts the stage finitely often. So $\mu\hat{}\infty$ is accessible infinitely often and since μ is initialised finitely often, $\mu\hat{}\infty$ is initialised finitely often. □

Lemma 7.15. Let ρ be a node working for requirement $P_{\Psi,k}$. Let s and $t > s$ be successive ρ-expansionary stages, and let $x \in \mathcal{C}_s(\rho)$. Suppose $z = \texttt{tr}_s(\rho, x) = \texttt{tr}_t(\rho, x)$ and let $u = \gamma_s(z)$. If $A_t \upharpoonright u \neq A_s \upharpoonright u$ or $W_t \upharpoonright u \neq W_s \upharpoonright u$, then $o^\rho_t(z) < o^\rho_s(z)$.

Proof. At stage $r \in (s, t)$, $\Gamma(A, W, z)[r] \uparrow$, then at the next τ-expansionary stage $r' \in (r, t]$ $\Gamma(A, W, z)[r'] = r'$. $\Gamma(A, W, z)[t] \geqslant \Gamma(A, W, z)[r']$ and $r' > \Gamma(A, W, z)[s]$, so $\Gamma(A, W, z)[t] \neq \Gamma(A, W, z)[s]$. s is a ρ-expansionary stage so $\Gamma(A, W, z)[s] = f^\rho_s(z)$, and t is also a ρ-expansionary stage so $\Gamma(A, W, z)[t] = f^\rho_s(t)$, then $f^\rho_s(z) \neq f^\rho_t(z)$, hence $o^\rho_t(z) < o^\rho_s(z)$. □

Lemma 7.16. Let ρ be a node working for requirement $P_{\Psi,k}$, then for all x only finitely many attacks are started for (ρ, x) while a particular z is the tracker for x.

Proof. Fix x; if z is the tracker of x for finitely many stages, then clearly only finitely many attacks are started for (ρ, x) while z is the tracker. So let z be the tracker for x for infinitely many stages. For an attack started at stage r, it is declared finished at stage s when we see $d^\rho_s(z) < d^\rho_r(z)$, so $o^\rho_s(z) < o^\rho_r(z)$. Note that (ρ, x) does not start another attack until there has been a ρ-expansionary stage after the previous attack was declared finished. Now if (ρ, x) starts infinitely many attacks while z is the tracker, then every one of these attacks started must be declared finished. Then we must see $o^\rho_s(z)$ decrease infinitely often, but this ordinal is from the ω^2-computable approximation of f^ρ so this cannot happen. Hence for all x only finitely many attacks are started for (ρ, x) while a particular z is the tracker for x. □

Lemma 7.17. Let ρ be a node working for requirement $P_{\Psi,k}$, and $x \in \mathcal{C}_s(\rho)$, then ρ enumerates finitely many numbers into A during each (ρ, x) attack.

Proof. Suppose not. Then there is a (ρ, x) attack such that ρ enumerates infinitely many numbers into A. Enumerations happen at ρ-expansionary stages, so by Lemma 7.15 we see the ordinal $o^\rho(z)$ decrease after each enumeration. Then if (ρ, x) enumerates infinitely many numbers into A we must see this ordinal decrease infinitely often, but this ordinal is from the ω^2-computable approximation of f^k so this cannot happen. Hence ρ enumerates finitely many numbers into A during each (ρ, x) attack. □

Lemma 7.18. Suppose τ is accessible infinitely often and initialised finitely often, then either $\tau\hat{}\infty$ or $\tau\hat{}\texttt{fin}$ is accessible infinitely often and initialised finitely often.

Proof. Suppose there are finitely many τ-expansionary stages. τ halts the stage only at τ-expansionary stages; hence τ halts the stage finitely often, so $\tau\hat{}\texttt{fin}$ is accessible infinitely often. τ is initialised finitely often and $\tau\hat{}\infty$ is accessible finitely often, so $\tau\hat{}\mathtt{fin}$ is initialised finitely often.

Now suppose there are infinitely many τ-expansionary stages but finitely many $\tau\hat{}\infty$ stages. τ halts the stage if it acts on a request or defines $\gamma^\rho(z)$ large during a (ρ, x) attack, where $z = \mathtt{tr}(x)$ and ρ is a daughter of τ.

Let t be the last stage $\tau\hat{}\infty$ was accessible. Since there are finitely many $\tau\hat{}\infty$ stages, finitely many sons ζ_k are visited, and there are finitely many $\zeta_k \uparrow$ stages for each son ever visited. A request is sent to τ only at $\zeta_k \uparrow$ stages, but there are finitely many of these so τ deals with finitely many requests; hence τ acts on a request finitely often.

Then since τ halts the stage infinitely often, $\gamma^\rho(z)$ large infinitely often during a (ρ, x) attack. Now since there are only finitely many $\tau\hat{}\infty$ stages, only finitely many daughters are ever visited; hence only finitely x are ever established. By Lemma 7.16, each of these only start finitely many attacks. Then there is a particular (ρ, x) attack where $\gamma^\rho(z)$ is defined large infinitely often. Now there are no ρ-expansionary stages after stage t, so after stage t, $\gamma^\rho(z)$ is defined large because there is some $(\hat{\rho}, \hat{x}) \in \mathtt{pro}(\rho, x)$ such that the computation $\Gamma_{\hat{\rho}}(A, W, \hat{z})$ has changed since the last τ-expansionary stage. Now $\mathtt{pro}(\rho, x)$ is only redefined after an enumeration into A; hence $\mathtt{pro}(\rho, x)$ does not change after stage t. Then there is a particular $(\hat{\rho}, \hat{x}) \in \mathtt{pro}(\rho, x)$ such that the computation $\Gamma_{\hat{\rho}}(A, W, \mathtt{tr}(\hat{\rho}, x))$ has changed since the last τ-expansionary stage infinitely often. Now the tracker for $\hat{x}$ can only change finitely often so there is some tracker $\hat{z}$ such that there are infinitely many A or W changes below $\gamma^{\hat{\rho}}(\hat{z})$. By Lemma 7.15, every time there is an A or W change below $\gamma(\hat{z})$ the ordinal $o^{\hat{\rho}}(\hat{z})$ must decrease. Now $\hat{\rho}\hat{}\infty \preceq \tau$, so $\Gamma_{\hat{\rho}}(A, W)$ is ω^2-c.a. But $o^{\hat{\rho}}(\hat{z})$ decreases infinitely often, so this is a contradiction.

Therefore τ halts the stage finitely often, so there are infinitely many $\tau\hat{}\infty$ stages. □

Lemma 7.19. Suppose ρ is accessible infinitely often and initialised finitely often, then either $\rho\hat{}\infty$ or $\rho\hat{}\mathtt{fin}$ is accessible infinitely often and initialised finitely often.

Proof. Suppose there are finitely many ρ-expansionary stages. Then ρ only halts the stage at ρ-expansionary stages, so ρ halts the stage finitely often; hence $\rho\hat{}\mathtt{fin}$ is accessible infinitely often. Since ρ is initialised finitely often and $\rho\hat{}\infty$ is accessible finitely often, then $\rho\hat{}\mathtt{fin}$ is initialised finitely often.

Now suppose there are infinitely many ρ-expansionary stages but there are only finitely many $\rho\hat{}\infty$ stages. New x are established only at $\rho\hat{}\infty$ stages, so only finitely many (ρ, x) are established. Let t be the last $\rho\hat{}\infty$ stage. Only μ such that $\mu \succ \rho\hat{}\infty$ can corrupt x; hence no x is corrupted after stage s. So a new tracker is only appointed if an x is uncorrupted; hence the tracker for each x changes at most once after stage t. By Lemma 7.16, each established x starts finitely many attacks. It follows from Lemma 7.17 that each (ρ, x) attack puts finitely many numbers into A. There are only finitely many established x and finitely many attacks for each x; hence ρ enumerates finitely many numbers into A. Therefore ρ halts the stage finitely often; thus there are infinitely many $\rho\hat{}\infty$ stages. □

Lemma 7.20. The true path is infinite.

Proof. Suppose there are only finitely many stages at which we define the collection of accessible nodes. Then since nodes are accessible finitely often, only finitely many (ρ, x) are ever established. After the last stage nodes are accessible no attack can be declared finished, so after this stage an attack can start for each (ρ, x) at most once. There are finitely many (ρ, x) so only finitely many attacks are ever started, but then the stage halts without defining the accessible nodes finitely often, a contradiction. So there are infinitely many stages at which we define the collection of accessible nodes. Then by Lemma 7.13, 7.14, 7.18 and 7.19, the true path is infinite. □

Lemma 7.21. Let ρ and $\hat{\rho}$ be nodes working for requirements $P_{\Psi,k}$ and $P_{\hat{\Psi},\hat{k}}$ respectively. Let $x \in \mathcal{C}_s(\rho)$ and $\hat{x} \in \mathcal{C}_s(\hat{\rho})$ with trackers z and $\hat{z}$ respectively. Suppose (ρ, x) is in an attack at stage s. If $(\hat{\rho}, \hat{x})$ is in an attack at stage s and $\hat{\rho}$ is to the right of $\rho\hat{\ }\infty$, then $\gamma_s^{\hat{\rho}}(\hat{z}) > \gamma_s^{\rho}(z)$.

Proof. Notice that every stage such that $\gamma^{\rho}(z)$ is defined large, $\hat{\rho}$ is intialised. Let t be the stage $\gamma^{\rho}(z)$ was last defined large. Then $\gamma_{t+1}^{\rho}(z) = \gamma_s^{\rho}(z)$. Now let r be the stage $(\hat{\rho}, \hat{x})$ was declared fully in an attack. $\hat{\rho}$ was initialised at stage t, so $r > t$; hence $\gamma_r^{\hat{\rho}}(\hat{z}) > \gamma_r^{\rho}(z) = \gamma_{t+1}^{\rho}(z)$. Now $\gamma_s^{\hat{\rho}}(\hat{z}) \geqslant \gamma_r^{\hat{\rho}}(\hat{z})$ and $\gamma_{t+1}^{\rho}(z) = \gamma_s^{\rho}(z)$; hence $\gamma_s^{\hat{\rho}}(\hat{z}) > \gamma_s^{\rho}(z)$. □

Lemma 7.22. Suppose y is established for π at stage s. Suppose a (ρ, x) attack causes a number to enter A below $\varphi_t(y)$ at stage $t > s$. If π is not initialised then $\rho \succeq \pi\hat{\ }\infty$.

Proof. Clearly if ρ is to the left of π then π gets initialised at stage t. Suppose $\rho\hat{\ }\infty \preceq \pi$ or ρ is to the right of π. Let r be the stage (ρ, x) is declared fully in an attack. Notice that π is not accessible between stages r and t. If y was not established at stage r then it will not be established until after stage t; hence $r > s$. $\varphi_t(y)\downarrow$ so it follows that $\varphi_r(y)\downarrow$ and $\varphi_t(y) = \varphi_r(y)$. At stage r, $\gamma_{r+1}(z)$ is defined with large use, so $\gamma_r(z) > \varphi_r(y)$. Since the use is non-decreasing it follows that $\gamma_t(z) > \varphi_t(y)$, a contradiction. Therefore if a (ρ, x) attack causes a number to enter A below $\varphi_t(y)$ at stage $t > s$ then $\rho \succeq \pi\hat{\ }\infty$. □

Lemma 7.23. Let π be a node working for requirement N_Φ and let ρ be a node working for requirement $P_{\Psi,k}$ with parent τ. Let z be the tracker for x. Suppose $\rho \succeq \pi\hat{\ }\infty$. Let r be a stage $\gamma_{r+1}^{\rho}(z)$ is defined to be large. If $\Phi(A, y)[r]\downarrow$ then for all stages $s > r$ during this attack $A_s \upharpoonright \varphi_r(y) = A_r \upharpoonright \varphi_r(y)$.

Proof. Let $s+1$ be the least counterexample. Then at stage s some $(\hat{\rho}, \hat{x})$ enumerated $\gamma_s^{\hat{\rho}}(\hat{z})$ into A and $\gamma_s^{\hat{\rho}}(\hat{z}) < \varphi_r(y)$. Notice that $(\hat{\rho}, \hat{x}) \neq (\rho, x)$ because $s+1$ is the least counterexample so $\varphi_r(y) = \varphi_s(y)$ and $\gamma_{r+1}^{\rho}(z) > \varphi_r(y)$. Now at stage s, (ρ, x) is still in its attack; hence either $\hat{\rho}$ is to the right of $\rho\hat{\ }\infty$ or $\hat{\rho}\hat{\ }\infty \preceq \rho$. If $\hat{\rho}\hat{\ }\infty \preceq \rho$ then by the assignment of requirements $\hat{\rho}\hat{\ }\infty \preceq \tau$. Thus the $(\hat{\rho}, \hat{x})$ attack started after stage r. If $\hat{\rho}$ is to the right of $\rho\hat{\ }\infty$ then $\hat{\rho}$ was initialised at stage r; thus the $(\hat{\rho}, \hat{x})$ attack started after stage r.

Let r' be the stage $(\hat{\rho}, \hat{x})$ was declared fully in an attack. At stage r' we define $\gamma_{r'+1}^{\hat{\rho}}(\hat{z})$ to be large. Since $r' > r$, $\gamma_{r'+1}^{\hat{\rho}}(\hat{z}) > \varphi_r(y)$. Since $\gamma^{\hat{\rho}}(\hat{z})$ is non-decreasing, so it follows that $\gamma_s^{\hat{\rho}} > \varphi_r(y)$, a contradiction. □

Lemma 7.24. Let π be a node working for requirement N_Φ and let ρ be a node working for requirement $P_{\Psi,k}$ with parent τ. Let z be the tracker for x and let s be the stage y was established. Suppose $\tau \succeq \pi\hat{\ }\infty$. Suppose (ρ, x) started an attack or enumerated a number into A at stage $r > s$. Then for all stages $t > r$ during this attack, if $\Gamma_\rho(A, W, z)[t]\downarrow$ then $\gamma_t(z) > \varphi_t(y)$.

Proof. Let r' be the next τ-expansionary stage after stage r. Then at stage r' we define $\gamma^\rho_{r'+1}(z)$ is defined to be large. Now $\tau \geq \pi\hat{\ }\infty$ and y is established; hence $\Phi(A, y)[r'] \downarrow$. Then by Lemma 7.23, for all stages $t > r'$ during this attack $A_t \upharpoonright \varphi_{r'}(y) = A_{r'} \upharpoonright \varphi_{r'}(y)$. Then $\varphi_{r'}(y) = \varphi_t(y)$. Now $\gamma^\rho_{r'+1}(z) > \varphi_{r'}(y)$; hence $\gamma^\rho_{r'+1}(z) > \varphi_t(y)$. $\gamma^\rho(z)$ is non-decreasing so it follows that $\gamma^\rho_t(z) > \varphi_t(y)$. Now between stages r and r', $\Gamma_\rho(A, W, z)[t] \uparrow$. Therefore we are done. □

Lemma 7.25. Suppose (ρ, x) is in an attack at stage s. Let $z = \mathtt{tr}_s(x)$ and $o^\rho_s(z) = \omega \cdot d^\rho_s(z) + b^\rho_s(z)$. Then during this attack, (ρ, x) enumerates at most $b^\rho_s(z)$ numbers into A after stage s.

Proof. Enumeration into A happens at ρ-expansionary stages so by Lemma 7.15, each time we enumerate a number into A during this attack we will see the ordinal $o^\rho_t(z)$ decrease. Let t be the first ρ-expansionary stage after (ρ, x) has enumerated $b^\rho_s(z)$ numbers into A since stage s. Then we have seen the ordinal $o^\rho_s(z)$ decrease $b^\rho_s(z)$ many times. So $o^\rho_t(z) \leqslant o^\rho_s(z) - b^\rho_s(z) = \omega \cdot d^\rho_s(z) + b^\rho_s(z) - b^\rho_s(z) = \omega \cdot (d^\rho_s(z) - n) + b^\rho_t(z)$ for some $n > 0$. Hence $d^\rho_t(z) < d^\rho_s(z)$, so the attack is declared finished and (ρ, x) will not enumerate any more numbers into A for this attack. □

Lemma 7.26. Let ρ and $\hat\rho$ be nodes working for requirements $P_{\Psi,k}$ and $P_{\hat\Psi,\hat k}$ respectively. Let $x \in \mathcal{C}_s(\rho)$ and $\hat x \in \mathcal{C}_s(\hat\rho)$ with trackers z and $\hat z$ respectively. Suppose x is corrupted, $(\hat\rho, \hat x)$ is in an attack at stage s, and $\hat\rho$ is to the right of $\rho\hat{\ }\infty$. Let r be the last ρ-expansionary stage. If $\gamma^{\hat\rho}_s(\hat z) < \gamma^\rho_s(z)$ then either x was corrupted at stage r or (ρ, x) was in an attack at stage r.

Proof. Let s be the least counterexample. Note that the last ρ-expansionary stage r is before the stage that the $(\hat\rho, \hat x)$ attack started, and x is neither corrupted nor uncorrupted during the $(\hat\rho, \hat x)$ attack as these actions would initialise $\hat\rho$. $\gamma^{\hat\rho}_s(\hat z) < \gamma^\rho_s(z)$, so by Lemma 7.21, (ρ, x) is not in an attack at stage s.

As in the argument for Lemma 7.21, if $\hat\rho$ was to the right of $\tau\hat{\ }\infty$ then $\gamma^{\hat\rho}_s(\hat z) > \gamma^\rho_s(z)$; hence $\hat\rho > \tau\hat{\ }\infty$. Let t be the stage the computation $\Gamma_{\hat\rho}(A, W, \hat z)[s]$ was defined. Note that this means $(A, W)_t \upharpoonright \gamma^{\hat\rho}_s(\hat z) = (A, W)_s \upharpoonright \gamma^{\hat\rho}_s(\hat z)$. Now if $\gamma^{\hat\rho}_s(\hat z)$ was defined to be large then $\gamma^{\hat\rho}_t(\hat z) > \gamma^\rho_t(z)$ because $\hat\tau > \tau\hat{\ }\infty$, but then it would follow that $\gamma^{\hat\rho}_s(\hat z) > \gamma^\rho_s(z)$; hence we did not define $\gamma^{\hat\rho}_s(\hat z)$ to be large at stage t. Then it was defined to be $\gamma^{\hat\rho}_{r_0}(\hat z)$ where r_0 is the last $\hat\tau$-expansionary stage before stage t.

Suppose $(A, W)_{r_0} \upharpoonright \gamma^\rho_{r_0}(z) \neq (A, W)_s \upharpoonright \gamma^\rho_{r_0}(z)$. Let $r_1 \in [r_0, s)$ be the least stage such that $(A, W)_{r_0} \upharpoonright \gamma^\rho_{r_0}(z) \neq (A, W)_{r_1+1} \upharpoonright \gamma^\rho_{r_0}(z)$.

Suppose $W_{r_0} \upharpoonright \gamma^\rho_{r_0}(z) \neq W_{r_1+1} \upharpoonright \gamma^\rho_{r_0}(z)$. Now (ρ, x) does not start an attack at this stage because it would initialise $\hat\rho$. If there is a W change while (ρ, x) is corrupted but we do not start an attack, then either x was corrupted at the last ρ-expansionary stage or (ρ, x) was in an attack at the last ρ-expansionary stage. Stage r was the last ρ-expansionary stage so the Lemma holds.

Now suppose $W_{r_0} \upharpoonright \gamma^\rho_{r_0}(z) = W_{r_1+1} \upharpoonright \gamma^\rho_{r_0}(z)$. Then it must be that $A_{r_0} \upharpoonright \gamma^\rho_{r_0}(z) \neq A_{r_1+1} \upharpoonright \gamma^\rho_{r_0}(z)$. Then this was cause by a (ρ', x') attack enumerating a number into A at stage r_1. Now this ρ' cannot be to the left of $\hat\rho$ as then $\hat\rho$ would be initialised at stage r_1. But then ρ' is to the right of $\rho\hat{\ }\infty$ and $r_1 < s$; hence the Lemma holds at stage r_1 and so either x was corrupted at stage r or (ρ, x) was in an attack at stage r, so we are done.

Now suppose $(A,W)_{r_0} \upharpoonright \gamma^{\rho}_{r_0}(z) = (A,W)_s \upharpoonright \gamma^{\rho}_{r_0}(z)$. If $\gamma^{\hat{\rho}}_s(\hat{z}) < \gamma^{\rho}_s(z)$ then $\gamma^{\hat{\rho}}_{r_0}(\hat{z}) < \gamma^{\rho}_{r_0}(z)$. s is the least counterexample so the Lemma holds at stage r_0; hence either x was corrupted at stage r or (ρ,x) was in an attack at stage r, so we are done. □

Lemma 7.27. Let ρ and $\hat{\rho}$ be nodes working for requirements $P_{\Psi,k}$ and $P_{\hat{\Psi},\hat{k}}$ respectively. Let $x \in \mathcal{C}_s(\rho)$ and $\hat{x} \in \mathcal{C}_s(\hat{\rho})$ with trackers z and $\hat{z}$ respectively. Suppose x is corrupted and $\rho\hat{\ }\infty \preceq \hat{\rho}$. If $(\rho,x) \notin \mathtt{pro}_s(\hat{\rho},\hat{x})$ and $(\hat{\rho},\hat{x})$ enumerates $\gamma^{\hat{\rho}}_s(\hat{z})$ into A at stage s then x is uncorrupted.

Proof. $\rho\hat{\ }\infty \preceq \hat{\rho}$ and $(\rho,x) \notin \mathtt{pro}_s(\hat{\rho},\hat{x})$ so x was established after the stage $\mathtt{pro}_s(\hat{\rho},\hat{x})$ was defined. x is corrupted so μ is to the right of $\hat{\rho}\hat{\ }\infty$. Then μ is initialised at every stage $\gamma^{\hat{\rho}}(\hat{z})$ is defined large. Let t be the last stage $\gamma^{\hat{\rho}}_{t+1}(\hat{z})$ is defined large. Now μ is initialised at stage t; hence q is appointed after stage t. $\mu > \rho\hat{\ }\infty$ so x is invented after stage t. Let r be the first τ-expansionary stage after x is invented, $\gamma^{\rho}_r(\mathtt{orig}(x))$ is defined large; hence $\gamma^{\rho}_r(\mathtt{orig}(x)) > \gamma^{\hat{\rho}}_{t+1}(\hat{z})$. Now $\gamma^{\rho}_r(\mathtt{orig}(x)) \leqslant \gamma^{\rho}_s(\mathtt{orig}(x))$ and $\gamma^{\hat{\rho}}_{t+1}(\hat{z}) = \gamma^{\hat{\rho}}_s(\hat{z})$ so it follows that $\gamma^{\hat{\rho}}_s(\hat{z}) < \gamma^{\rho}_s(\mathtt{orig}(x))$. Therefore x is uncorrupted. □

Lemma 7.28. Let π be a node working for requirement N_{Φ}. Let r be the stage y is established. Let ρ be a node working for requirement $P_{\Psi,k}$. Suppose $\tau\hat{\ }\infty \preceq \pi\hat{\ }\infty \preceq \rho$. Suppose either x was declared fully corrupted at stage $t > r$ or (ρ,x) was declared to have fully finished an attack at stage $t > r$. If (ρ,x) wants to attack at stage $s > t$ then $\Phi_{\pi}(A,y)[s]\downarrow$.

Proof. Let s be the least counterexample. Let $s_0 < s$ be the last $\pi\hat{\ }\infty$ stage. Then since $\Phi_{\pi}(A,y)[s]\uparrow$, some $(\hat{\rho},\hat{x})$ enumerated a number into A at stage $t_1 \in [s_0,s)$ such that $\gamma^{\hat{\rho}}_{t_1}(\hat{z}) < \varphi_{t_1}(y)$. Notice that by Lemma 7.22, $\hat{\rho} \geq \pi\hat{\ }\infty$; hence $t_1 = s_0$. $\tau\hat{\ }\infty \preceq \pi\hat{\ }\infty \preceq \rho$ so by the assignment of requirements, $\hat{\tau} \succeq \rho\hat{\ }\infty$ or $\hat{\rho}$ is to the right of $\rho\hat{\ }\infty$.

In the latter case, stage t is a ρ-expansionary stage so $\hat{\rho}$ is initialised at stage t. Let t_0 be the stage that $(\hat{\rho},\hat{x})$ attack starts. Now $\hat{x}$ was established after stage t; hence $\hat{x}$ was declared fully corrupted at stage $t' \in (r,t_0)$. By s is the least counterexample and $s > t_0$ so it follows that $\Phi_{\pi}(A,y)[t_0]\downarrow$. Then by Lemma 7.23, $\gamma^{\hat{\rho}}_{t_1}(\hat{z}) > \varphi_{t_1}(y)$, a contradiction.

Then $\hat{\tau} \succeq \rho\hat{\ }\infty$. By Lemma 7.24, y was established after $(\hat{\rho},\hat{x})$ was declared fully in an attack. Then the $(\hat{\rho},\hat{x})$ was declared fully in an attack before stage t. Now suppose we are in the case where x was declared fully corrupted at stage x at stage t. Note that the stage $(\hat{\rho},\hat{x})$ is declared fully in an attack is a ρ-expansionary stage; hence x was corrupted during the $(\hat{\rho},\hat{x})$ attack. Then it follows that μ is to the right of $\hat{\rho}$. Therefore x was established after the last stage $\gamma^{\hat{\rho}}(\hat{z})$ was defined large; hence $(\rho,x) \notin \mathtt{pro}_{t_1}(\hat{\rho},\hat{x})$. Then by Lemma 7.27, x is uncorrupted at stage t_1. Therefore (ρ,x) does not want to attack at stage s, a contradiction.

Now consider the case that (ρ,x) was declared to have fully finished an attack at stage t. Now notice that $\hat{\tau}$ is not accessible during the (ρ,x) attack; hence $\gamma^{\hat{\rho}}(\hat{z})$ was last defined large before the (ρ,x) attack starts. Let t_2+1 be the stage the (ρ,x) attack starts. Now consider the following cases:

Case 1. $(\rho,x) \in \mathtt{pro}_{t_2}(\hat{\rho},\hat{x})$. Then by Lemma 7.7, $\gamma^{\hat{\rho}}_{t_2}(\hat{z}) > \gamma^{\rho}_{t_2}(z)$. But at stage t_2+1 there is a W below $\gamma^{\rho}_t(z)$ prompting the start of an attack. But this change is also below $\gamma^{\hat{\rho}}_{t_2}(\hat{z})$ and causes the use to be lifted large at the next $\hat{\tau}$-expansionary stage. But then the use $\gamma^{\hat{\rho}}(\hat{z})$ is lifted large after the (ρ, x) attack and hence also after y is established. Therefore $\gamma^{\hat{\rho}}_{t_1}(\hat{z}) > \varphi_{t_1}(y)$, a contradiction.

Case 2. $(\rho, x) \notin \texttt{pro}_{t_1}(\hat{\rho}, \hat{x})$. Then by Lemma 7.27, x is uncorrupted at stage t_1. Therefore (ρ, x) does not want to attack at stage s, a contradiction. □

Lemma 7.29. Every N_Φ requirement is met.

Proof. Fix Φ. The true path is infinite and every infinite path on the tree of strategies has a node working for requirement N_Φ; let π be the node on the true path working for N_Φ. If $\Phi(A)$ is not total then we are done, so suppose $\Phi(A)$ is total. Then we need to show $\Phi(A)$ that is ω-c.a. by defining an ω-computable approximation. At each $\pi\hat{}\infty$ stage we establish a new input y by giving it an ordinal as follows. Let s first $\pi\hat{}\infty$ stage after y is established, then define

$$o^\Phi_s(y) = n + \sum_{(\rho,x)\in\mathcal{B}(y)} b^\rho_s(x)$$

where: $\mathcal{B}(y)$ is the set of (ρ, x) such that $x \in \mathcal{C}_s(\rho)$ is fully corrupted and $\tau\hat{}\infty \preceq \pi\hat{}\infty \preceq \rho$; n is the number of ρ such that (ρ, x) is in an attack at stage s, $\Gamma_\rho(A, W, \texttt{tr}(x))[s]\downarrow$ and the parent of ρ, τ, is such that $\tau > \pi\hat{}\infty$.

By Lemma 7.24, the only (ρ, x) with $\tau > \pi\hat{}\infty$ that can injure $\Phi(A, y)$ are in an attack at stage s and $\Gamma_\rho(A, W, \texttt{tr}(x))[s]\downarrow$. Also by Lemma 7.24, such ρ can only injure $\Phi(A, y)$ at most once. At stage s there are n such (ρ, x) in an attack and each of these can only injure $\Phi(A, y)$ at most once; hence $\Phi(A, y)$ can be injured by such enumerations at most n many times.

By Lemma 7.22, the only (ρ, x) that can injure $\Phi(A, y)$ extend $\pi\hat{}\infty$. We have just dealt with the case where $\tau > \pi\hat{}\infty$, so all that is left is the case where $\tau\hat{}\infty \preceq \pi$. Now suppose at stage $t > s$, x is declared fully corrupted or (ρ, x) is declared to have fully finished an attack. By Lemma 7.28, if (ρ, x) wants to attack at a stage $t' > t$, then $\Phi(A, y)[t']\downarrow$; then by Lemma 7.23, it follows that for all stages r during this attack $\gamma^\rho_r(\texttt{tr}_r(x)) > \varphi_r(y)$.

Then the only (ρ, x) with $\tau\hat{}\infty \preceq \pi\hat{}\infty \preceq \rho\hat{}\infty$ that can injure $\Phi(A, y)$ at stage t are such that x was declared fully corrupted at a stage $r < s$; hence they are in the set $\mathcal{B}(y)$. After this attack is declared finished, by Lemma 7.28, for all stages r during any attack started after this stage $\gamma^\rho_r(\texttt{tr}_r(x)) > \varphi_r(y)$. Therefore, by Lemma 7.25, each $(\rho, x) \in \mathcal{B}(y)$ can injure $\Phi(A, y)$ at most $b^\rho_s(z)$ times.

So the ordinal defined above is the maximum possible number of enumerations into A below $\phi(y)$. Therefore this is the maximum number of changes to the computation $\Phi(A, y)$. Hence we have an ω-computable approximation for $\Phi(A)$ as desired. □

Lemma 7.30. Let η be a node working for requirement R_Θ. If $\Theta(A, W)$ is total and η is on the true path then $\Delta_\eta(A, W)$ is total.

Proof. $\Theta(A, W)$ is total so there are infinitely many η-expansionary stages. At an η-expansionary we leave $\Delta_\Theta(A, W, p)$ undefined only if p is an anchor for some μ and we appoint a new follower, but by Lemma 7.12 for each p this happens finitely often. So $\limsup_s \Delta_\Theta(A, W)[s]$ goes to infinity. Now we need to check that the use is bounded for all p. Suppose p is eventually not an anchor for any daughter of η. Then $\Delta_\Theta(A, W, p)$ is eventually defined with use 0; hence the use of such p is bounded. So now suppose p is an anchor for μ, a daughter of η, and $p = \texttt{ac}_s(\mu)$ for infinitely many stages. Now by Lemma 7.12, there are only finitely many followers appointed while p is the anchor. Let q be the last follower appointed and let t be the stage q is appointed. Then if $\Delta_\Theta(A, W, p)[s]$ is defined at stage $s > t$, its use is defined to be $\theta_s(q)$. $\theta_s(q)$ has bounded use because $\Theta(A, W)$ is total. Therefore $\Delta_\Theta(A, W, p)$ has bounded use for all p; hence $\Delta_\Theta(A, W)$ is total. □

Lemma 7.31. Let τ be a node working for requirement P_Ψ. If $\Psi(A, W, Q)$ is total and τ is on the true path then $\Gamma_\Psi(A, W)$ is total.

Proof. Since $\Psi(A, W, Q)$ is total, there are infinitely many τ-expansionary stages. At each τ-expansionary stage we defined a longer initial segment of $\Gamma_\Psi(A, W)$, so $\limsup_s \operatorname{dom} \Gamma_\Psi(A, W,)[s]$ goes to infinity. If z is eventually not a tracker of any x for any daughter of τ then $\Gamma_\Psi(A, W, z)$ is defined with use 0, so the use of such z is bounded. Now consider z which is the tracker for some x for some daughter of τ, ρ, for infinitely many stages. Suppose there is a stage t such that for all $s > t$, (ρ, x) is not in an attack at stage s. A computation $\Gamma_\Psi(A, W, z)$ defined at stage $s > t$ is defined with use $\psi_s^\rho(I_s(\rho, x))$. Since z is never cancelled, ρ is initialised finitely often, so by Lemma 7.11 the interval $I(x)$ must stabilise. $\Psi(A, W, Q)$ is total so the use $\psi_s(x')$ for every $x' \in I(x)$ is bounded; hence the use of $\Gamma_\Psi(A, W, z)$ is bounded for such z. So now consider that there is no such stage t, then by Lemma 7.16 it must be that there is some attack that is never finished. During an attack a number is enumerated into A at every ρ-expansionary stage, then by Lemma 7.17 there are finitely many ρ-expansionary stages. So let t be the last ρ-expansionary stage. Now $\texttt{pro}_s(\rho, x)$ is only redefined after an A enumeration by (ρ, x), but there are no ρ-expansionary stages after stage t; hence for all $s, r > t$, $\texttt{pro}_s(\rho, x) = \texttt{pro}_r(\rho, x)$. Now for $(\hat{\rho}, \hat{x}) \in \texttt{pro}_s(\rho, x)$, $\hat{\rho}\hat{\ }\infty \prec \tau$, so $\hat{\rho}\hat{\ }\infty$ is also initialised finitely often. Then the tracker for $(\hat{\rho}, \hat{x})$ is only change finitely often; let $\hat{z}$ be the last tracker. $\hat{\rho}\hat{\ }\infty$ is also accessible infinitely often; hence for all $(\hat{\rho}, \hat{x}) \in \texttt{pro}_s(\rho, x)$, $\Gamma_{\hat{\rho}}(A, W)$ is ω^2-c.a., so the computation $\Gamma_{\hat{\rho}}(A, W, \texttt{tr}_s(\hat{\rho}, \hat{x}))$ changes finitely often. For each $(\hat{\rho}, \hat{x}) \in \texttt{pro}_s(\rho, x)$, $\hat{z}$ is the last tracker so $\hat{x}$ is never uncorrupted while $\hat{z}$ is the tracker. Then the use is defined to be large due to the uncorruption of some protected $\hat{x}$ finitely often. Then it follows that there is a stage r such that at every stage $s > r$ where we define the computation $\Gamma_\Psi(A, W, z)[s]$, we define it with use $\gamma_r(z)$. Hence the use of $\Gamma_\Psi(A, W, z)$ is bounded for all z, and therefore $\Gamma_\Psi(A, W)$ is total. □

Remark 7.32. If there is no node on the true path that works for requirement $P_{\hat{\Psi}}$, then there is some node α working for requirement P_Ψ or R_Θ on the true path such that every son and daughter of α has been assigned a node on the true path. Recall that we say a parent is been closed below the infinite outcome of a daughter or the divergent outcome of a son. Also recall that we stop placing children on the tree once the parent has been closed. Then if α has infinitely many sons and daugters on the true path then the finite outcome of every daughter of α and the convergent outcome of every son of α is on the true path. Without loss of generality, let α be a node working for requirement P_Ψ. Then it follows that $\Psi(A, W, Q)$ is total. Then by Lemma 7.31, $\Gamma_\Psi(A, W)$ is total. Since the finite outcome of every daughter of α is on the true path, $\Gamma_\Psi(A, W)$ is not ω^2-c.a.; then $A \oplus W$ is not totally ω^2-c.a. This means that for all functionals Ψ and Θ, requirements R_Θ and P_Ψ are met immediately, so we are done.

So suppose $A \oplus W$ is totally ω^2-c.a. Then every parent node is eventually closed; hence there is a node on the true path working for every P_Ψ and R_Θ requirement. The following lemmas will use this assumption.

Lemma 7.33. Every R_Θ requirement is met.

Proof. Fix Θ. $A \oplus W$ is totally ω^2-c.a. and the true path is infinite, so there is a node η on the true path working for R_Θ. If $\Theta(A, W)$ is not total then we are done. So suppose $\Theta(A, W)$ is total. Then $\eta\hat{}\infty$ is on the true path, and by Lemma 7.30 $\Delta_\Theta(A, W)$ is total. Since $A \oplus W$ is totally ω^2-c.a. and $\Delta_\Theta(A, W)$ is total, $\Delta_\Theta(A, W)$ is ω^2-c.a. Then since the true path is infinite there must be a daughter of η, μ, such that $\mu\hat{}\infty$ is on the true path. Then μ is initialised finitely often; hence it has finitely many anchors, so let p be the last anchor. By Lemma 7.12 only finitely many followers are appointed, so let q be the last follower appointed.

If $\Theta(A, W, q) = 0$, then there is a μ-expansionary stage s such that $\Theta(A, W, q)[s] = 0$ after q is appointed. q is the last follower, so there is a stage where μ will either enumerate q into Q_{s+1} or send a request. If this request was cancelled at stage r then it is cancelled because $\Delta_\eta(A, W, p)[r]\uparrow$. Let t be the next η-expansionary stage after the request was cancelled. If there was some $(\rho, x) \in \mathtt{pro}_t(\mu)$ such that $q < \psi_t^\rho(I_t(x))$ then a new follower is appointed, but q is the last follower so this does not happen. Then another request to enumerate q into Q will be sent. $\Delta_\Theta(A, W)$ is total so we will only cancel the request finitely many times. So there is a request sent that is never cancelled. Then μ is not accessible until the request is carried out. There are infinitely many μ-expansionary stages; hence the request is eventually carried out, so $q \in Q$.

$\Theta(A, W, q)[s] = 0$ prompting the enumeration of q into Q. Then since $\Theta(A, W, q) = 1$, there is some stage $t > s$ such that either $A_s \upharpoonright \theta_s(q) \neq A_t \upharpoonright \theta_s(q)$ or $W_s \upharpoonright \theta_s(q) \neq W_t \upharpoonright \theta_s(q)$. By Lemma 7.1, $\delta_s(p) \geqslant \theta_s(q)$, so it is also the case that either $A_s \upharpoonright \delta_s(p) \neq A_t \upharpoonright \delta_s(p)$ or $W_s \upharpoonright \delta_s(p) \neq W_t \upharpoonright \delta_s(p)$. So there is an η-expansionary stage after stage $r > s$ where $\Delta_\Theta(A, W, p)[r]\uparrow$ and $q \in Q_r$; this prompts the appointment of a new follower, but this contradicts that q is the last follower appointed. Therefore if $\Theta(A, W, q) = 1$ then $q \notin Q$.

So we have shown that diagonalisation is successful, $\Theta(A, W) \neq Q$; hence requirement R_Θ is met. □

Lemma 7.34. Every P_Ψ requirement is met.

Proof. Fix Ψ. $A \oplus W$ is totally ω^2-c.a. and the true path is infinite, so there is a node τ on the true path working for R_Ψ. If $\Psi(A, W, Q)$ is not total then we are done, so suppose $\Psi(A, W, Q)$ is total. Then by Lemma 7.31, $\Gamma_\Psi(A, W)$ is total. $A \oplus W$ is totally ω^2-c.a. so there is a daughter of τ, ρ, such that $\rho\hat{}\infty$ is on the true path. Since $\rho\hat{}\infty$ is on the true path, every x is eventually established, so now we show that there is an ω^2-computable approximation for $\Psi(A, W, Q)$.

Now we define $\alpha_t(x) = \omega \cdot m_t(x) + k_t(x)$. ρ is initialised finitely often so consider the stages after the last initialisation. Now let s be the first ρ-expansionary stage after x is established, and let $z_0 = \mathtt{tr}_s(\rho, x) = \mathtt{orig}(\rho, x)$. Then define $m_s(x) = d_s^\rho(z_0) + 2$ and $k_s(x) = b_s^\rho(z_0)$. For stages r where x is not corrupted define $m_{r+1}(x) = m_r(x)$ and $k_{r+1}(x) = b_r^\rho(z_0)$.

Suppose x is corrupted. Let t be the first ρ-expansionary stage after corruption and let z_1 be the new tracker. Then define $m_{t+1}(x) = m_t(x) - 1$ and $k_{t+1}(x) = d_t^\rho(z_1)$.

If x is uncorrupted. Let t be the first ρ-expansionary stage after uncorruption, and define $m_t(x) = m_{t-1}(x) - 1$ and $k_t(x) = b^\rho_t(z_0)$.

For stages r while x is corrupted define $m_{r+1}(x) = m_r(x)$ and $k_{r+1}(x) = d^\rho_r(z_1)$.

Now define

$$o^\Psi_s(x) = \sum_{y \leqslant x} \alpha_s(y)$$

Suppose x has not been taken over. While x uses its original tracker it just follows the ordinal of its tracker. By Lemma 7.4, if a number enters Q below the use $\psi(I(x))$ then x is either taken over or declared corrupted and by Lemma 7.5 x is taken over if another number enters Q below $\psi(I(x))$. Therefore while x uses its original tracker there are no numbers entering Q; hence this ordinal works while x is not corrupted. Now if x gets corrupted or uncorrupted then we need to decrease $m(x)$ by one. By Lemma 7.5 we will only need to do this a maximum of two times hence $m_s(x) = d^\rho_s(z_0) + 2$ gives us enough room to count for these changes.

Now suppose x is corrupted. Note that we declare a new ordinal at the next ρ-expansionary stage, so if there are multiple changes between ρ-expansionary stages we only need to account for one of these changes. Also note that we do not need to worry about any changes during a (ρ, x) attack because wait until the first ρ-expansionary stage after the attack is finished before declaring a new ordinal.

If there is an A change below $\psi(I(x))$ at some stage t while (ρ, x) is not in an attack then by Lemma 7.3 **(4)** and Lemma 7.7 this must come from the attack of some $(\hat\rho, \hat x)$ such that $(\rho, x) \notin \mathtt{pro}_t(\hat\rho, \hat x)$. Now it is the case that either $\hat\rho \succ \rho\hat{\ }\infty$ or $\hat\rho$ is to the right of $\rho\hat{\ }\infty$. In the latter case, it follows from Lemma 7.26 that either x was corrupted at stage r or (ρ, x) was in an attack at stage r, where r is the last ρ-expansionary stage before the computation $\Gamma_\rho(A, W, \mathtt{tr}(x))[s]$ was defined. Then we do not need to declare a new ordinal until the next ρ-expansionary stage; hence we can charge the decrease to the case that holds. In the case that $\hat\rho \succ \rho\hat{\ }\infty$ then since $(\rho, x) \notin \mathtt{pro}_t(\hat\rho, \hat x)$, by Lemma 7.27, when $(\hat\rho, \hat x)$ enumerates a number into A at stage t, x is uncorrupted.

By Lemma 7.5, if there is a Q change below $\psi(I(x))$ while x is corrupted then x is taken over by some $x' < x$.

If there is a W change below $\psi(I(x))$, then we start an attack for (ρ, x) at some stage t. Since there are infinitely many ρ-expansionary stages this attack eventually finishes. Then there is a stage r such that $d^\rho_r(z_1) < d^\rho_t(z_1)$, and since $k_t(x) = d^\rho_t(z_1)$, we have $k_r(x) < k_t(x)$; hence we see a decrease in $\alpha(x)$.

Suppose x has been taken over by some x'. Now if there is a change below $\psi(x)$ then there is a change below $\psi(I(x'))$. Therefore by the above argument, we see a decrease in $\alpha(x')$. x can only be taken over by $x' < x$ and $\alpha(x')$ has been included in $o^\psi(x)$ for $x' < x$; hence if x is taken over and there is a change below $\psi(x)$ then there is a decrease in $o^\psi(x)$ as required.

Therefore this definition of o^ψ gives us an ω^2-computable approximation for $\Psi(A, W, Q)$. □

References

[1] Bahareh Afshari, George Barmpalias, S. Barry Cooper, and Frank Stephan. Post's programme for the Ershov hierarchy. *J. Logic Comput.*, 17(6):1025–1040, 2007.

[2] Klaus Ambos-Spies, Rodney G. Downey, and Martin Monath. On the sacks splitting theorem and approximating computations. In Preparation.

[3] Klaus Ambos-Spies, Nan Fang, Nadine Losert, Wolfgang Merkle, and Martin Monath. Array noncomputability: a modular approach. In Preparation.

[4] Klaus Ambos-Spies, Carl G. Jockusch, Jr., Richard A. Shore, and Robert I. Soare. An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees. *Trans. Amer. Math. Soc.*, 281(1):109–128, 1984.

[5] Klaus Ambos-Spies and Nadine Losert. Universally array noncomputable sets. In Preparation.

[6] Katherine Arthur. Maximality in the α-c.a. degrees. Master's thesis, Victoria University of Wellington, 2016.

[7] Katherine Arthur, Rodney G. Downey, Noam Greenberg, and Daniel Turetsky. The distribution of maximality in the totally α-c.a. degrees. Submitted.

[8] George Barmpalias. Hypersimplicity and semicomputability in the weak truth table degrees. *Arch. Math. Logic*, 44(8):1045–1065, 2005.

[9] George Barmpalias, Rodney G. Downey, and Noam Greenberg. Working with strong reducibilities above totally ω-c.e. and array computable degrees. *Trans. Amer. Math. Soc.*, 362(2):777–813, 2010.

[10] Mark Bickford and Chris F. Mills. Lowness properties of r.e. sets. Manuscript, UW Madison, 1982.

[11] Paul Brodhead, Rod Downey, and Keng Meng Ng. Bounded randomness. In *Computation, physics and beyond*, volume 7160 of *Lecture Notes in Comput. Sci.*, pages 59–70. Springer, Heidelberg, 2012.

[12] John Chisholm, Jennifer Chubb, Valentina S. Harizanov, Denis R. Hirschfeldt, Carl G. Jockusch, Jr., Timothy McNicholl, and Sarah Pingrey. Π^0_1 classes and strong degree spectra of relations. *J. Symbolic Logic*, 72(3):1003–1018, 2007.

[13] P. Cholak, R. Downey, L. Fortnow, E. Gasarch, W. Kinber, M. Kummer, S. Kurtz, and T. Slaman. Degrees of inferability. In *Conference on Computational Learning Theory*, pages 180–192, 1992.

[14] Peter Cholak, Rodney Downey, and Michael Stob. Automorphisms of the lattice of recursively enumerable sets : Promptly simple sets. *Trans. Amer. Math. Soc.*, 332:555–570, 1992.

[15] Peter Cholak, Rodney G. Downey, and Stephen Walk. Maximal contiguous degrees. *J. Symbolic Logic*, 67(1):409–437, 2002.

[16] G. Downey, Rodgey and Noam Greenberg. A hierarchy of computably enumerable turing degrees. *Bull. Symb. Log.*, 24:53–89, 2018.

[17] Rodney Downey and Richard Shore. Degree theoretical definitions of the low$_2$ recursively enumerable sets. *Journal of Symbolic Logic*, 60:727–756, 1995.

[18] Rodney G. Downey. A note on btt-degrees. *Rendiconti Seminario Matematico Dell'Universita e Del Politecnico di Torino*, 58:449–456, 2000.

[19] Rodney G. Downey. Some computability-theoretical aspects of reals and randomness. In *The Notre Dame Lectures*, Lecture Notes in Logic, pages 97–146. Association for Symbolic Logic, 2005.

[20] Rodney G. Downey and Jr. Carl Jockusch. t-degrees, jump classes and strong reducibilities. *Trans. Amer. Math. Soc.*, 301:103–136, 1987.

[21] Rodney G. Downey and Noam Greenberg. Totally $< \omega^\omega$ computably enumerable and m-topped degrees. In *Theory and applications of models of computation*, volume 3959 of *Lecture Notes in Comput. Sci.*, pages 46–60. Springer, Berlin, 2006.

[22] Rodney G. Downey and Noam Greenberg. *A Hierarchy of Turing Degrees*. Annals of Mathematics Studies. Princeton University Press, 2020.

[23] Rodney G. Downey, Noam Greenberg, and Rebecca Weber. Totally ω-computably enumerable degrees and bounding critical triples. *J. Math. Log.*, 7(2):145–171, 2007.

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

[25] Rodney G. Downey, Denis R. Hirschfeldt, André Nies, and Frank Stephan. Trivial reals. In *Proceedings of the 7th and 8th Asian Logic Conferences*, pages 103–131, Singapore, 2003. Singapore Univ. Press.

[26] Rodney G. Downey, Carl G. Jockusch, Jr., and Michael Stob. Array nonrecursive sets and multiple permitting arguments. In *Recursion theory week (Oberwolfach, 1989)*, volume 1432 of *Lecture Notes in Math.*, pages 141–173. Springer, Berlin, 1990.

[27] Rodney G. Downey, Carl G. Jockusch, Jr., and Michael Stob. Array nonrecursive degrees and genericity. In *Computability, enumerability, unsolvability*, volume 224 of *London Math. Soc. Lecture Note Ser.*, pages 93–104. Cambridge Univ. Press, Cambridge, 1996.

[28] Rodney G. Downey and Steffen Lempp. Contiguity and distributivity in the enumerable Turing degrees. *J. Symb. Logic*, 62(4):1215–1240, 1997.

[29] Rodney G. Downey and Keng Meng Ng. Splitting into degrees with low computational strength. *Annals of Pure and Applied Logic*, 169:803–834, 2018.

[30] Rogney G. Downey. Lattice nonembeddings and initial segments of the recursively enumerable. *Ann. Pure and Appl. Logic*, 49:97–119, 1990.

[31] Y Ershov. Positive equivalences. *Algebra i Logika*, 10:620–650, 1971.

[32] Yuri L. Ershov. A certain hierarchy of sets. I. *Algebra i Logika*, 7(1):47–74, 1968.

[33] Yuri L. Ershov. A certain hierarchy of sets. II. *Algebra i Logika*, 7(4):15–47, 1968.

[34] Yuri L. Ershov. A certain hierarchy of sets. III. *Algebra i Logika*, 9:34–51, 1970.

[35] Richard Friedberg. Two recursively enumerable sets of incomparible degrees of unsolvability. *Proc. Natl. Acad. Sci. USA*, 43:236–238, 1957.

[36] Noam Greenberg and Dan Turetsky. *Bulletin of Symbolic Logic*, 24:147–164, 2018.

[37] Leo Harrington and Robert I. Soare. Post's program and incomplete recursively enumerable sets. *Proc. Nat. Acad. Sci. U.S.A.*, 88(22):10242–10246, 1991.

[38] Li Ling Ko. PhD thesis.

[39] Martin Kummer and Marcus Schaeffer. Cuppability of simple and hypersimple sets. *Notre Dame Journal of Formal Logic*, 48:349–369, 2007.

[40] Stuart A. Kurtz. Notions of weak genericity. *Journal of Symbolic Logic*, 48:764–770, September 1983.

[41] A. H. Lachlan. Embedding nondistributive lattices into the recursively enumerable degrees. In Wilfred Hodges, editor, *Conference in Mathematical Logic*, volume Lecture Notes in Mathematics No. 255, pages 149–177. Springer-Verlag, 1972.

[42] A. H. Lachlan. Two theorems on the many one degrees of recursively enumerable sets. *Algebra i Logika*, 11:216–229, 1972.

[43] Alistair H. Lachlan and Robert I. Soare. Not every finite lattice is embeddable in the recursively enumerable degrees. *Adv. in Math.*, 37(1):74–82, 1980.

[44] Steffen Lempp and Manuel Lerman. A finite lattice without critical triple that cannot be embedded into the enumerable Turing degrees. *Ann. Pure Appl. Logic*, 87(2):167–185, 1997. Logic Colloquium '95 Haifa.

[45] Steffen Lempp, Manuel Lerman, and D. Reed Solomon. Embedding finite lattices into the computably enumerable degrees—a status survey. In *Logic Colloquium '02*, volume 27 of *Lect. Notes Log.*, pages 206–229. Assoc. Symbol. Logic, La Jolla, CA, 2006.

[46] M. Lerman. *Degrees of Unsolvability*. Perspectives in Mathematical Logic. Springer-Verlag, Heidelberg, 1983. 307 pages.

[47] Donald A. Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Z. Math. Logik Grundlagen Math.*, 12:295–310, 1966.

[48] Michael McInerney. *Topics in Algorithmic Randomness and Computability Theory*. PhD thesis, Victoria University of Wellington, 2016.

[49] Michael McInerney and Keng Meng Ng. Separating weak α-change generic and α-change generic. Submitted.

[50] McInerney Michael and Keng Meng Ng. Multiple genericity i: Bounding theorems and downward density. *Israel J. Math.*, page accepted.

[51] Martin Monath. *On Array Noncomputable Degrees, Maximal Pairs and Simplicity Properties*. PhD thesis, University of Heidelberg, 2020.

[52] A. A. Muchnik. On the unsolvability of the problem of reducibility in the theory of algorithms. N. S. 108:194–197, 1956.

[53] André Nies. Lowness properties and randomness. *Adv. Math.*, 197(1):274–305, 2005.

[54] André Nies. Reals which compute little. In *Logic Colloquium '02*, volume 27 of *Lect. Notes Log.*, pages 261–275. Assoc. Symbol. Logic, La Jolla, CA, 2006.

[55] André Nies, Richard A. Shore, and Theodore A. Slaman. Interpretability and definability in the recursively enumerable degrees. *Proc. London Math. Soc. (3)*, 77(2):241–291, 1998.

[56] André Nies, Frank Stephan, and Sebastiaan A. Terwijn. Randomness, relativization and Turing degrees. *J. Symbolic Logic*, 70(2):515–535, 2005.

[57] Emil L. Post. Recursively enumerable sets of positive integers and their decision problems. *Bull. Amer. Math. Soc.*, 50:284–316, 1944.

[58] Joseph R. Shoenfield. *Mathematical logic*. Association for Symbolic Logic, Urbana, IL, 2001. Reprint of the 1973 second printing.

[59] Richard A. Shore. Natural definability in degree structures. In *Computability theory and its applications (Boulder, CO, 1999)*, volume 257 of *Contemp. Math.*, pages 255–271. Amer. Math. Soc., Providence, RI, 2000.

[60] R. I. Soare. The infinite injury priority method. *Journal of Symbolic Logic*, 41:513–530, 1976.

[61] R. I. Soare. *Recursively Enumerable Sets and Degrees*. Springer, New York, 1985.

[62] Robert I. Soare. Automorphisms of the lattice of recursively enumerable sets I: maximal sets. *Annals of Math.*, 100:80–120, 1974.

[63] Alan Mathison Turing. On computable numbers with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc. (2)*, 42:230–265, 1936. A correction, 43:544–546.

[64] Yongge Wang. *Randomness and Complexity*. PhD thesis, University of Heidelberg, 1996.

[65] Barry Weinstein. *On Embedding of the Lattice 1-3-1 into the Recursively Enumerable Degrees*. PhD thesis, University of California, Berkeley, 1988.

School of Mathematics and Statistics, Victoria University, P.O. Box 600, Wellington, New Zealand
E-mail address: `Rod.Downey@vuw.ac.nz`

School of Mathematics and Statistics, Victoria University, P.O. Box 600, Wellington, New Zealand
E-mail address: `greenberg@msor.vuw.ac.nz`

School of Mathematics and Statistics, Victoria University, P.O. Box 600, Wellington, New Zealand
E-mail address: `ellen.hammatt@vuw.ac.nz`