Maximality in the α-c.a. Degrees

by

Katherine Arthur

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Master of Science
in Mathematics.

Victoria University of Wellington
2016

Abstract

In [4], Downey and Greenberg define the notion of totally α-c.a. for appropriately small ordinals α, and discuss the hierarchy this notion begets on the Turing degrees. The hierarchy is of particular interest because it has already given rise to several natural definability results, and provides a definable antichain in the c.e. degrees. Following on from the work of [4], we solve problems which are left open in the aforementioned relating to this hierarchy. Our proofs are all constructive, using strategy trees to build c.e. sets, usually with some form of permitting. We identify levels of the hierarchy where there is absolutely no collapse above any totally α-c.a. c.e. degree, and construct, for every $\alpha \leqslant \varepsilon_0$, both a totally α-c.a. c.e. minimal cover and a chain of totally α-c.a. c.e. degrees cofinal in the totally α-c.a. c.e. degrees in the cone above the chain's least member.

Acknowledgments

I would first like to acknowledge my late father, whose final gift allowed me to return to my studies sooner than I'd hoped. I would also like to thank Robin, my mother, and my grandmother, for their valiant efforts as my personal cheerleading squad.

I have had the distinct benefit of not one, but two exemplary supervisors. It is with gratitude that I mention Noam and Rod for their kind encouragement, endless patience and sage guidance, and my particular thanks to Noam for generously funding my Masters year.

Contents

Chapter 1

Introduction

This thesis follows on from Downey and Greenberg's recent collaborative work titled *A transfinite hierarchy of lowness notions in the computably enumerable degrees, unifying classes, and natural definability* ([4]). In the authors' own words, [4] is written in furtherance of "[understanding] the relationship between dynamic properties of sets and functions and their algorithmic complexity". In pursuit of this goal, [4] introduces a new hierarchy on the Turing degrees based on the relative complexity of approximations to the functions in those degrees. The contribution of this thesis, which is joint work with Downey and Greenberg, is to provide answers to specific questions which arise from the aforementioned hierarchy, particularly in relation to the c.e. degrees, and which are left open by [4]. The overarching goal is to understand the extent to which the hierarchy collapses, and to identify unusual or interesting features.

In this thesis, we prove a number of new results about this hierarchy. We prove the following theorems (which will be explained by formal definitions in the sections to follow):

(1) for $\alpha < \varepsilon_0$, above any α-c.a. c.e. degree there is no collapse in level α^ω and above;

(2) above any ω-c.a. c.e. degree there is a c.e. degree which is totally ω^3-c.a. but not totally ω-c.a.;

(3) there is no collapse (in any level) above any superlow c.e. degree;

(4) there is a pair of c.e. degrees $\boldsymbol{a} < \boldsymbol{d}$ such that $\boldsymbol{d}$ is totally ω-c.a., and every totally ω-c.a. c.e. degree above $\boldsymbol{a}$ is bounded by $\boldsymbol{d}$;

(5) there is a totally ω-c.a. c.e. degree which is bounded by no maximal totally ω-c.a. c.e. degree; and,

(6) for $\alpha \leqslant \varepsilon_0$, every c.e. degree which is not totally α-c.a. bounds a non-uniformly totally α-c.a. degree.

Though the details are not given explicitly in this thesis, results (2), (4) and (5) can be generalised with minor modification to their respective proofs. Thus, we further assert that:

(2)* for $n \geqslant 1$, above any ω^n-c.a. c.e. degree there is a degree which is totally ω^{n+2}-c.a. and not totally ω^n-c.a.;

for $\alpha \leqslant \varepsilon_0$ a power of ω,

(4)* there is a pair of c.e. degrees $\boldsymbol{a} < \boldsymbol{d}$ such that $\boldsymbol{d}$ is totally α-c.a., and every totally α-c.a. c.e. degree above $\boldsymbol{a}$ is bounded by $\boldsymbol{d}$; and,

(5)* there is a totally α-c.a. c.e. degree which is bounded by no maximal totally α-c.a. c.e. degree.

Not only do we solve these questions, but in doing so we introduce new techniques for their solution. For instance, we have the first example of a construction of one of these bounded classes where there is infinitary positive activity along the true path of the strategy tree (see Theorem 2.1.3). We believe that our techniques will have wider applications. To clarify the meaning and implications of the stated theorems above, we now discuss the terms and clarify the context from which the theorems arise.

1.1. The Mind-Change Function

To explain the new hierarchy and its origins, we begin with the Shoenfield Limit Lemma [11]. This states that a function f is computable from $\emptyset'$ if, and only if, f has a *computable approximation*; that is, there is a uniformly computable sequence $\langle f_s \rangle$ such that, for all x, $\lim_s f_s(x) = f(x)$. It is clear that for every c.e. degree $\boldsymbol{d}$, every function $f \in \boldsymbol{d}$ has a computable approximation. We then have the *mind-change function* of $\langle f_s \rangle$ (on x), which is just as one might guess: $m_{\langle f_s \rangle}(x) = \#\{s \mid f_s(x) \neq f_{s+1}(x)\}$, the number of times the approximation $\langle f_s \rangle$ 'changes its mind' on value x. We can compare functions, and the degrees that contain them, by the relative complexity of functions that bound their respective mind change functions. This method of comparison and classification forms the basis of the new hierarchy.

The mind-change function is central to the definition of the class of *array computable* degrees, as defined by Downey, Jockusch and Stob [6] [7]. A c.e. Turing degree $\boldsymbol{a}$ is array computable if every function $f \in \boldsymbol{a}$ has a computable approximation $\langle f_s \rangle$ such that for all n there are at most n-many stages s for which $f_s(n) \neq f_{s+1}(n)$. In other words, for every function $f \in \boldsymbol{a}$, the mind-change function of $\langle f_s \rangle$ is bounded by the identity function. The class of array computable degrees is particularly noteworthy for an uncommon characteristic it possesses: it captures the combinatorics of a number of constructions. A c.e. degree is array noncomputable if and only if it contains a perfect, thin Π^0_1 class [2], if and only if it contains a c.e. set with maximal Kolmogorov complexity [9], if and only if it does not have a strong minimal cover in the Turing degrees [8] [1]. There are further equivalences in addition to these which are detailed in [4] and cement the natural definability of the class of array noncomputable degrees.

[1] It's not important what exactly these terms are; we intend to convey that the class of array noncomputable degrees is a ubiquitous class.

A minor but consequential variation on the definition of the array computable degrees was put forth by J. Miller (unpublished, 2005). We call a function ω-*computably approximable*, or ω-c.a., if it has a computable approximation for which the mind change function is bounded by a computable function. This turns out to be equivalent to being weak truth-table reducible to $\emptyset'$. We then say that a c.e. degree is *totally* ω*-c.a.* if every function in the degree is ω-c.a.. Downey, Greenberg and Weber [5] showed that the class of totally ω-c.a. degrees (referred to there as totally ω*-c.e.*, and sometimes elsewhere as totally ω*-r.e.*) do in fact capture the dynamics of some constructions, providing natural definability results. Notably, the class of not totally ω-c.a. degrees is precisely that of the c.e. degrees which bound a (weak) critical triple, and that of the c.e. degrees which bound a (weak) wtt triple. Yet another equivalence is presented in [4], relating the class of not totally ω-c.a. degrees to presentations of left-c.e. reals, and several other equivalences (listed in [4]) have been proven by various authors [2]. It is also of note that maximal totally ω-c.a. degrees exist, forming a naturally definable antichain in the c.e. degrees. The only other such known antichain is formed by the maximal contiguous degrees [3].

1.2. $\mathscr{R}$-c.a. Functions

Given the promise shown by the class of totally ω-c.a. degrees, we want to take this idea further in the hopes of obtaining yet more natural definability results and otherwise noteworthy results (such as definable antichains). The approach taken in [4] is to extend, or generalise, the notion of "(totally) ω-c.a.". Downey and Greenberg begin with the notion of $\mathscr{R}$-c.a. functions for a computable well-ordering $\mathscr{R}$, and then identify an appropriate association of ordinals to computable well-orderings in order to produce a meaningful

[2]The intent in listing these equivalences is the same as behind those given for the array noncomputable degrees: the specifics are not important, just the fact that the equivalences exist.

definition of α-c.a. functions and, by extension, totally α-c.a. degrees.

DEFINITION 1.2.1: Let $\mathscr{R} = \langle R, <_R \rangle$ be a computable well-ordering on a computable set R. An *$\mathscr{R}$-computable approximation* of $f : \omega \to \omega$ is a computable approximation $\langle f_s \rangle_{s<\omega}$ of f, equipped with a uniformly computable sequence $\langle o_s \rangle_{s<\omega}$ of functions from ω into R such that for all x and all s:

(i) $o_{s+1}(x) \leqslant_R o_s(x)$; and,

(ii) if $f_{s+1}(x) \neq f_s(x)$, then $o_{s+1}(x) <_R o_s(x)$.

Given that $\mathscr{R}$ is a well-ordering, $\langle o_s(x) \rangle_{s<\omega}$ cannot be an infinite descending sequence for any x. As such, $\{s \mid f_s(x) \neq f_{s+1}(x)\}$ is finite, and $\lim_s f_s(x)$ certainly exists. We then refer to a function $f : \omega \to \omega$ as *$\mathscr{R}$-computably approximable*, or *$\mathscr{R}$-c.a.*, if it has an $\mathscr{R}$-computable approximation.

Similarly, we define a *partial $\mathscr{R}$-computable approximation* of a function $f : \omega \to \omega$ to be a computable approximation $\langle f_s \rangle_{s<\omega}$ of f, equipped with a uniformly computable sequence $\langle o_s \rangle_{s<\omega}$ of partial functions such that:

(i) for all s, dom o_s is an initial segment of ω, and dom $o_s \subseteq$ dom o_{s+1};

(ii) $\langle \text{dom } o_s \rangle_{s<\omega}$ is uniformly computable;

(iii) for all s, and all $x \notin \text{dom } o_s$, $f_s(x) = 0$; and,

(iv) for all s and all $x \in \text{dom } o_s$, $o_{s+1}(x) \leqslant_R o_s(x)$, and if $f_{s+1}(x) \neq f_s(x)$ then $o_{s+1}(x) <_R o_s(x)$.

We say that a list $\langle \langle f^e_s, o^e_s \rangle_{s<\omega} \rangle_{e<\omega}$ of partial $\mathscr{R}$-computable approximations is *effective* if the functions f^e_s are computable, uniformly in both e and s, the functions o^e_s are partial computable, uniformly in both e and s, and their domains dom o^e_s are computable, uniformly in both e and s. The following proposition is proved in [4]:

Proposition 1.2.2: *There is an effective list $\langle\langle f_s^e, o_s^e\rangle_{s<\omega}\rangle_{e<\omega}$ of partial $\mathscr{R}$-computable approximations such that, letting $f^e = \lim_s f_s^e$, the list $\langle f^e\rangle_{e<\omega}$ contains every $\mathscr{R}$-c.a. function.*

There is no uniform listing of total $\mathscr{R}$-computable approximations for all $\mathscr{R}$-c.a. functions, so we will need to rely on this effective list of partial approximations instead. However, if the context requiring the list is able to accommodate the addition of a new, terminal element to (the order type of) $\mathscr{R}$, we can produce an effective list $\langle\langle f_s^e, o_s^e\rangle_{s<\omega}\rangle_{e<\omega}$ of total $(\mathscr{R}+1)$-computable approximations by altering the partial approximations given by Proposition 1.2.2. [3] The list $\langle f^e\rangle_{e<\omega}$, where $f^e = \lim_s f_s^e$, then contains every $\mathscr{R}$-c.a. function.

A naïve attempt to define the α-c.a. functions might be to let f be α-c.a. if and only if it is $\mathscr{R}$-c.a. for some computable well-ordering $\mathscr{R}$ of order type α. However, it is a result of Ershov that every Δ_2^0 function is $\mathscr{R}$-c.a. for some computable well-ordering $\mathscr{R}$ of order type ω. It is clearly no good to us to build a hierarchy on the c.e. degrees if it will only have one level, so we dismiss this approach.

Another suggestion is to fix some Π_1^1 path through Kleene's $\mathcal{O}$, and restrict the computable well-orderings we consider to those given by notations on that path, but this has its own problems. Firstly, though the path may be cofinal in $\mathcal{O}$ and thus provide a notation for every computable ordinal, it will not exhaust every Δ_2^0 function. Secondly, the choice of any such path would be arbitrary, as there is no canonical way to select one. We would then have different hierarchies for different choices of path, which is substantially less than ideal. What we really seek, or indeed require for the notion of "α-c.a." to be at all useful, is invariance.

The solution found by Downey and Greenberg is to restrict attention to a

[3] Let $\langle\langle g_s^e, m_s^e\rangle_{s<\omega}\rangle_{e<\omega}$ be given by Proposition 1.2.2. Then, for all e, s, x, we define $f_s^e(x) = g_s^e(x)$. If $m_s^e(x)\uparrow$, define $o_s^e(x)$ to be the new terminal element; otherwise define $o_s^e(x) = m_s^e(x)$.

particularly well-behaved class of computable well-orderings. For computable well-orderings $\mathscr{R}$ and $\mathscr{S}$, if $\mathscr{R}$ and $\mathscr{S}$ are computably isomorphic then a function is $\mathscr{R}$-c.a. if, and only if, it is $\mathscr{S}$-c.a.. Hence, it is stipulated that all members of this well-behaved class which are of the same length should also be computably isomorphic. To achieve canonicity, it is also required that all (reasonable) associated functions such as successor, predecessor etc. are computable. Downey and Greenberg discovered that, up to ordinal ε_0, the Cantor normal form encapsulates the required information to produce the sought-after class of ordinals.

1.3. The Cantor Normal Form

Let α be an ordinal; α can be uniquely expressed as the sum

$$\alpha = \omega^{\alpha_1} \cdot n_1 + \omega^{\alpha_2} \cdot n_2 + \cdots + \omega^{\alpha_k} \cdot n_k$$

where $\alpha_1 > \alpha_2 > \cdots > \alpha_k$ are ordinals, and $n_1, n_2, \ldots, n_k$ are non-zero natural numbers. This is the *Cantor normal form* [1] of α. We denote

$$\varepsilon_0 = \sup\{\omega, \omega^\omega, \omega^{\omega^\omega}, \omega^{\omega^{\omega^\omega}}, \ldots\},$$

the least ordinal γ such that $\omega^\gamma = \gamma$. For all ordinals $\alpha < \varepsilon_0$, every ordinal appearing in the Cantor normal form of α is strictly less than α.

Let $\mathscr{R} = \langle R, <_R \rangle$ be a computable well-ordering, and let $|\cdot| : R \to \mathrm{otp}(\mathscr{R})$ be the unique isomorphism between $\mathscr{R}$ and its order type. We define the *Cantor normal form function* $\mathrm{nf}_{\mathscr{R}}$ on $\mathscr{R}$ by letting

$$\mathrm{nf}_{\mathscr{R}}(z) = \langle (z_1, n_1), (z_2, n_2), \ldots, (z_k, n_k) \rangle \text{ for } z \in R,$$

where each $z_i \in R$, $z_1 >_R z_2 >_R \cdots >_R z_k$, and $n_1, n_2, \ldots, n_k$ are nonzero natural numbers, and

$$|z| = \omega^{|z_1|} \cdot n_1 + \omega^{|z_2|} \cdot n_2 + \cdots + \omega^{|z_k|} \cdot n_k.$$

DEFINITION 1.3.1: A computable well-ordering $\mathscr{R}$ is *canonical* if its associated Cantor normal form function $\mathrm{nf}_{\mathscr{R}}$ is also computable.

We note that if the relations of ordinal addition and exponentiation by ω in $\mathscr{R}$ are computable, then $\mathscr{R}$ is canonical. It is shown in [4] that there is a canonical, computable well-ordering of order type ε_0. Further, if $\mathscr{R}$ is a canonical, computable well-ordering, then $\mathscr{R} \upharpoonright z$ (the initial segment of $\mathscr{R}$ up to $z \in R$) is also a canonical, computable well-ordering. Hence, for every $\alpha \leqslant \varepsilon_0$, there must be a canonical, computable well-ordering of order type α. We now have the well-behaved class of well-orderings required to produce a meaningful definition of α-c.a. functions for $\alpha \leqslant \varepsilon_0$.

DEFINITION 1.3.2: Let $\alpha \leqslant \varepsilon_0$. A function f is *α-computably approximable* if it is $\mathscr{R}$-c.a. for some (all) canonical well-ordering $\mathscr{R}$ of order type α.

We then say that a Turing degree $\boldsymbol{d}$ is *totally α-c.a.* if every function $f \in \boldsymbol{d}$ is α-c.a.; then, equivalently, $\boldsymbol{d}$ is totally α-c.a. if and only if every function $f \leqslant_{\mathrm{T}} \boldsymbol{d}$ is α-c.a.. Further, if $\boldsymbol{d}$ is totally α-c.a., then it is low_2.

Application of Proposition 1.2.2 to a canonical, computable well-ordering $\mathscr{R}$ of order type α will produce an effective list $\langle f_s^e, o_s^e \rangle$ of partial α-computable approximations such that, letting $f^e = \lim_s f_s^e$, the list $\langle f^e \rangle_{e<\omega}$ contains *every* α-c.a. function. If we fix a canonical well-ordering $\mathscr{R}_{\varepsilon_0}$ of order type ε_0, we can, uniformly in $\alpha < \varepsilon_0$, fix an effective list $\langle f_s^{\alpha,e}, o_s^{\alpha,e} \rangle$ of partial α-computable approximations.

It is prudent that the class of ω-c.a. functions by this definition be precisely that of the ω-c.a. functions by Miller's definition. Let f be a function with an $\mathscr{R}$-computable approximation $\langle f_s, o_s \rangle$ for canonical well-ordering $\mathscr{R}$ of order type ω. For all x, define $h(x) = o_0(x)$; then the mind-change function for $\langle f_s \rangle$ is bounded by computable h. Conversely, let f be a function that has a computable approximation $\langle f_s \rangle$ for which the mind-change function is bounded by a computable function h. For all x, define $o_0(x) = h(x)$. For any stage $s > 0$, if $f_s(x) = f_{s-1}(x)$, define $o_s(x) = o_{s-1}(x)$; otherwise, define $o_s(x) = o_{s-1}(x) - 1$. Then $\langle f_s, o_s \rangle$ is an ω-computable approximation of f. Hence the two notions are equivalent, as required.

1.4. The Totally α-c.a. Hierarchy

Let $\gamma < \alpha \leqslant \varepsilon_0$; it is fairly clear that any totally γ-c.a. degree is also totally α-c.a.. It is then natural to ask whether the set of totally γ-c.a. degrees is in fact a proper subset of the totally α-c.a. degrees. The following theorem from [4] confirms this to be the case when α is a power of ω.

Theorem 1.4.1: *Let $\alpha \leqslant \varepsilon_0$. There is a totally α-c.a. degree which is not totally γ-c.a. for any $\gamma < \alpha$ if, and only if, α is a power of ω. If α is a power of ω, then in fact there is a c.e. degree which is totally α-c.a. but not totally γ-c.a. for any $\gamma < \alpha$.*

Hence, there is a totally ω^2-c.a. degree which is not totally ω-c.a., a totally ω^3-c.a. degree which is not totally ω^2-c.a., and so forth. This forms the first picture of the hierarchy of totally α-c.a. degrees; it collapses (above $\mathbf{0}$) only between powers of ω. What is not addressed by this theorem is whether the hierarchy exhibits further collapse in the cone above or below any given c.e. degree.

Let $\alpha \leqslant \varepsilon_0$ be a power of ω. We say that there is no collapse above degree $\boldsymbol{a}$ in level α of the hierarchy if there is a degree $\boldsymbol{d} \geqslant \boldsymbol{a}$ which is totally α-c.a. but not totally γ-c.a. for any $\gamma < \alpha$. That is, that level of the hierarchy contains at least one member (not contained in any lower level) when restricted to the cone above $\boldsymbol{a}$. Recall the beginning of this chapter where we stated the theorems proved in this thesis; their meaning should now be mostly clear.
First, we examine collapse in the hierarchy. In Section 2.1 we introduce the notion of *maximal totally α-c.a.* degrees, and prove that each level α^ω and above contains a maximal member in the cone above any totally α-c.a. c.e. degree (for $\alpha < \varepsilon_0$). The existence of these maximal members implies no

collapse in these levels. Then in Section 2.2, above a given totally ω-c.a. c.e. degree $\boldsymbol{a}$ we build a degree $\boldsymbol{d} > \boldsymbol{a}$ for which every $f \leqslant_T \boldsymbol{d}$ is ω^3-c.a. but there is a $g \leqslant_T \boldsymbol{d}$ which is not ω-c.a.. This is our only construction which does not use some form of permitting to build a c.e. set. In Section 2.3 we prove that there is a maximal member in every level above a superlow c.e. degree (which is necessarily totally ω-c.a.), implying no further collapse in any level of the hierarchy above such a degree. There are still situations in which collapse (or lack thereof) remains to be proved, particularly in the ω^2 level above any (non-superlow) totally ω-c.a. c.e. degree.

We then turn our attention to interesting features of the hierarchy. In Section 2.4, we construct a local 'minimal cover' that bounds all totally ω-c.a. c.e. degrees above a constructed c.e. degree $\boldsymbol{a}$. We then extend this in Theorem 2.4.9 into a chain of totally ω-c.a. c.e. degrees cofinal in the totally ω-c.a. c.e. degrees above a constructed c.e. degree $\boldsymbol{a}$. Finally, in Section 2.5 we introduce the notion of *uniformly totally α-c.a.*, and use non-totally α-c.a. permitting to construct a totally, but not uniformly totally, α-c.a. c.e. degree bounded by a given not totally α-c.a. c.e. degree.

1.5. Strategy Trees

Each construction in this text is a priority construction that employs a strategy tree to meet an infinite set of *requirements*. Each element, or *node* of the tree, is a finite sequence of symbols. We describe the strategy tree by specifying the following:

(i) An association of requirements to nodes. We then say that a node *works* for the requirement with which it is associated. Often, all nodes of the same level will work for the same requirement, though this is not always the case.

(ii) For each node that works for a requirement, all possible *outcomes*.

From this base, we are able to define the tree recursively. We begin with the fact that the empty node is always on the tree. If we have established that a node σ is on the tree, and σ is associated with requirement R, then the immediate successor nodes of σ on the tree are of the form $\sigma\hat{\ }o$, where o is an outcome of nodes working for R.

We specify a linear ordering, denoted $<$, on the collection of outcomes of any node. We then say that outcome o is *stronger* than outcome o' if $o < o'$; on the tree, outcome o will then be to the left of outcome o'. We proceed to extend this ordering to the entire tree; for nodes σ, τ we say $\sigma < \tau$ if:

(i) $\sigma \prec \tau$, in which case we refer to σ as being *stronger* than τ; OR,

(ii) there are η, o, o' such that $\sigma \succeq \eta\hat{\ }o$ and $\tau \succeq \eta\hat{\ }o'$, where $o < o'$; we then say σ *lies to the left of* τ.

If $\sigma < \tau$ by either case above, τ is described as *weaker* than σ.

The construction then describes, for each stage s, the collection γ_s of nodes *accessible* at s. We may choose to leave γ_s empty at a stage s if the construction warrants it, but if γ_s is non-empty we assume it to contain at least the empty node. For the constructions in this text, for each s at which γ_s is non-empty, there is a node σ on the tree such that γ_s comprises $\{\eta \mid \eta \preceq \sigma\}$, the downward closure of σ.

We say that a node σ lies on the *true path* γ_ω of the tree if $\sigma \in \gamma_s$ for infinitely many s, and the same cannot be said for any node τ which lies to the left of σ. The true path γ_ω is linearly ordered under $\preceq$; it will need to be proved in each construction that γ_ω is infinite and thus contains, for each requirement, an associated node which should ensure the requirement's satisfaction.

Often, at the conclusion of a stage, we will *initialise* specified nodes. To initialise a node is to *cancel*, or remove, any current parameters (such as followers) associated with the node. We ensure that, if we initialise a node τ at s, every node weaker than τ is also initialised at s. We say the construction is *fair* to a node τ if τ is only initialised finitely often; this is a crucial quality for nodes on the true path. We refer to a follower or computation as *perma-*

nent if, after the stage of its definition, it is never cancelled or destroyed.
We use notation and conventions consistent with those in [4].

Chapter 2

The New Theorems

2.1. Maximal Totally α-c.a. Degrees

Let $\alpha \leqslant \varepsilon_0$, and let $\boldsymbol{d}$ be a Turing degree which is totally α-c.a.. We refer to $\boldsymbol{d}$ as *maximal totally α-c.a.* if there is no $\boldsymbol{a} > \boldsymbol{d}$ such that $\boldsymbol{a}$ is totally α-c.a.. It is established in [4] that maximal totally α-c.a. c.e. degrees do exist for every $\alpha \leqslant \varepsilon_0$ which is a power of ω (and thus exist in every level of the hierarchy), in addition to the following theorem:

Theorem 2.1.1: *Let $\beta < \varepsilon_0$. Every totally ω^β-c.a. c.e. degree is bounded by a strictly greater totally $\omega^{\beta+1}$-c.a. c.e. degree.*

Corollary 2.1.2: *Let $\alpha \leqslant \varepsilon_0$ be a power of ω, let $\boldsymbol{a}$ be a maximal totally α-c.a. c.e. degree. Then $\boldsymbol{a}$ is not totally γ-c.a. for any $\gamma < \alpha$.*

Proof. Suppose instead that $\boldsymbol{a}$ is totally γ-c.a. for some $\gamma < \alpha$. By Theorem 2.1.1, there is a c.e. degree $\boldsymbol{d} > \boldsymbol{a}$ which is totally $\gamma \cdot \omega$-c.a.. Since $\gamma \cdot \omega \leqslant \alpha$, $\boldsymbol{d}$ is also totally α-c.a., a contradiction to the maximality of $\boldsymbol{a}$.

$\square$

Given Corollary 2.1.2, to show that there is no collapse in level $\alpha \leqslant \varepsilon_0$ above a c.e. degree $\boldsymbol{a}$, it is sufficient to show that there is a maximal totally α-c.a. degree $\boldsymbol{d}$ such that $\boldsymbol{d} > \boldsymbol{a}$. Our first new theorem shows that for any $\alpha < \varepsilon_0$, any totally α-c.a. c.e. degree $\boldsymbol{a}$ is bounded by a maximal totally β-c.a. degree $\boldsymbol{d}$ for $\beta \geqslant \alpha^\omega$. Thus, the cone above $\boldsymbol{a}$ does not exhibit any collapse of the hierarchy in the levels α^ω and above.

THEOREM 2.1.3: *Let $\alpha < \varepsilon_0$ be a power of ω. Let $\boldsymbol{a}$ be a c.e. degree which is totally α-c.a., and let $\beta \geqslant \alpha^\omega$ also be a power of ω. There is then a maximal totally β-c.a. c.e. degree $\boldsymbol{b}$ such that $\boldsymbol{b} > \boldsymbol{a}$.*

Let α, β, $\boldsymbol{a}$ be fixed. Given $\langle A_s \rangle_{s<\omega}$, a computable enumeration of some $A \in \boldsymbol{a}$, we proceed to build a c.e. set D with the intent that $\deg_{\mathrm{T}}(A \oplus D)$ is the required $\boldsymbol{b}$. For convenience, we may safely assume that for every e, j, s, if j enters W_e at stage s, then all k such that $j \leqslant k < s$ enter W_e at s.

Requirements:

Let $\langle \Phi_e \rangle_{e<\omega}$ be an enumeration of all consistent functionals. To ensure that $\deg_{\mathrm{T}}(A \oplus D)$ is totally β-c.a., we must meet the set of requirements given by:

$$\text{for all } e < \omega,\ Q_e : \text{If } \Phi_e(A, D) \text{ is total, then it is } \beta\text{-c.a.}$$

We must also ensure the maximality of $\deg_{\mathrm{T}}(A \oplus D)$; we achieve this by requiring, for all $e < \omega$, that either $W_e \leqslant_{\mathrm{T}} A \oplus D$ or $\deg_{\mathrm{T}}(A \oplus D \oplus W_e)$ is not totally β-c.a. To this end, we enumerate a Turing functional Λ_e with the intent that either $W_e \leqslant_{\mathrm{T}} A \oplus D$, or $\Lambda_e(D, W_e)$ is not β-c.a. Normally we would build $\Lambda_e(A, D, W_e)$, but in this case we discover that A is not required as an oracle of Λ.
By Proposition 1.2.2, let $\langle \langle f^i_s, o^i_s \rangle_{s<\omega} \rangle_{i<\omega}$ be an effective list of partial β-computable approximations such that letting $f^i = \lim_s f^i_s$, the sequence $\langle f^i \rangle_{i<\omega}$ contains every β-c.a. function. We may assume that any β-c.a. function f appears in this sequence as some f^i, and that for this approximation $\langle f^i \rangle$, we have $\bigcup_s \operatorname{dom} o^i_s = \omega$. As there is no uniform listing of total

β-computable approximations for all β-c.a. functions, we use this latter assumption to identify appropriate candidates to work with in the sequence $\langle f^i \rangle_{i<\omega}$.
With the preceding in mind, we meet the following set of requirements:

$$\text{for all } e, i < \omega,\ P_e^i : \text{If } \bigcup_s \operatorname{dom} o_s^i = \omega, \text{ either } \Delta_e^i(A, D) = W_e, \text{ or } \Lambda_e(D, W_e) \neq f^i.$$

Discussion:

To meet the P and Q requirements, we build a strategy tree.

Let τ be a node working for Q_d. To meet Q_d, for every $x < \omega$, τ must *certify* the computation $\Phi_d(A, D, x)$ by appointing an ordinal (below β) when we first see $\Phi_d(A, D, x) \downarrow$. In order to produce a β-computable approximation, this ordinal should correctly bound the 'number of times' that the computation $\Phi_d(A, D, x)$ will be destroyed.

To meet a requirement P_e^i in isolation, a node σ working for P_e^i selects a follower $p = p(\sigma, j)$. Whenever $f_s^i(p) = \Lambda_e(D, W_e, p)[s]$, we then enumerate $\lambda_{e,s}(p)$ into D_{s+1}. However, this change in D may destroy a computation $\Phi_d(A, D, x)[s]$, and could thus be problematic to a bound appointed by a node working for Q_d.
Suppose at a stage s we first see $\Phi_d(A, D, x) \downarrow$, and $o_s^i(p) \downarrow$ for follower $p = p(\sigma, j)$ of σ working for P_e^i. Then we are immediately able to give an ordinal bound (the actual value $o_s^i(p)$) on σ acting for p, and we can allow $\Phi_d(A, D, x)$ to be injured by p. We do so by ensuring that the value of $o_s^i(p)$ is taken into account when deciding the bound appointed to the $\Phi_d(A, D, x)$ computation.
If instead $o_s^i(p) \uparrow$, we cannot give a bound at this stage on action for p. Because of this uncertainty, we must not allow p to injure $\Phi_d(A, D, x)$. If we appoint p and do not define $\lambda_e(p)$ in that same stage, it is possible that p is never cancelled and σ is never again visited, so Λ_e could end up partial;

hence, we must define $\lambda_e(p)$ immediately when p is appointed. If τ certifies a computation $\Phi_d(A, D, x)$ after p is appointed (and $\lambda_e(p)$ defined) but before we see $o^i_s(p)\downarrow$, then $\lambda_e(p)$ may be too small to avoid injuring $\Phi_d(A, D, x)$. We cannot allow τ to wait for $o^i(p)\downarrow$ before appointing a bound to the computation, as $\langle f^i_s, o^i_s\rangle_{s<\omega}\rangle_{i<\omega}$ is a list of partial approximations, so we cannot guarantee that this will ever occur. However, if we are able to redefine the use $\lambda_{e,t+1}(p)$ to be large at a stage at which $\Phi_d(A, D, x)\downarrow[t]$, before allowing σ to act for p, we could prevent injury to the computation.

Before we see $o^i(p)\downarrow$, σ will certainly not need to act for p; this is because σ would act for p when we see $f^i_s(p) = \Lambda_e(D, W_e, p)[s]$, but $f^i_s(p)\downarrow$ implies $o^i_s(p)\downarrow$. Hence, we can wait until we see $o^i(p)\downarrow$ to redefine the use. Then, if $W_e \not\leq_T D$, we use $j \downarrow W_e$ to permit σ to increase the use $\lambda_e(p)$ to be large, allowing σ to attack with p. As we cannot be sure that any given follower will be able to be permitted in this manner, we continue to appoint followers to σ while none is permitted.

We note that the stage at which we see $j \downarrow W_e$ may not be a stage at which σ is accessible. We need to define $\lambda_e(p)$ to ensure that $\Lambda_e(D, W_e)$ is total, and we cannot be sure that σ will become accessible thereafter; hence, we need to act on permissions regardless of what nodes are accessible at that stage. We only allow a follower p to be permitted and act if $\Phi_d(A, D, x)\downarrow$ for all $\langle d, x\rangle$ such that p may not injure $\Phi_d(A, D, x)$. The lifted use $\lambda_e(p)$ is then larger than that of all computations p may not injure. We use Δ to ensure this by defining $\Delta_\sigma(A, D, j)$ to have the same use as the maximum use of any protected computation $\Phi_d(A, D, x)$ of p. If $\Delta_\sigma(A, D, j)\downarrow$ when σ next becomes accessible after p's permission, we can be certain that all protected computations converged at the stage at which p became permitted.

Since A is beyond our control, we also need to be aware of the manner in which enumeration into A can affect the requirements. Of course, it is possible that σ has a permitted follower p, and at some later stage there is an enumeration into A that destroys a computation $\Phi_d(A, D, x)$ that p may not injure. In this case, we cannot guarantee that $\lambda_e(p)$ is still greater than $\varphi_d(x)$ when the computation $\Phi_d(A, D, x)$ converges again, and we cannot lift the use of p (by permission from W_e) a second time. The follower p would then be in a position to cause injury to $\Phi_d(A, D, x)$, so must be cancelled when σ is next accessible to prevent unacceptable injury. It is possible that infinitely many followers will be appointed to σ only to be cancelled in response to enumeration into A; in this case we show that $\Delta_\sigma(A, D) = W_e$. As a technical note, this proof is unusual in that it allows this potentially infinitary action at a node on the true path.

To keep track of change in A, we build a shadow functional $\hat{\Phi}_d(A)$ such that if $\Phi_d(A, D)$ is total, then $\hat{\Phi}_d(A)$ is total. We call $c < \omega$ the *tracker* of $x < \omega$ if c is the input for which we define $\hat{\Phi}_d(A, c)$ to have use $\varphi_e(x)$. Let $\langle\langle g^i_s, m^i_s\rangle_{s<\omega}\rangle_{i<\omega}$ be an effective list of partial α-computable approximations such that letting $g^i = \lim_s g^i_s$, the sequence $\langle g^i\rangle_{i<\omega}$ contains every α-c.a. function. We know that $\boldsymbol{a}$ is totally α-c.a.; hence there is an i such that $\hat{\Phi}_d(A) = g^i$. As the strategy progresses, we guess this index i (correctly) using the fact that A is low$_2$. This is a Σ^0_3 guessing process, requiring infinitely many outcomes. On finding the correct i, the value $m^i(c)$ is a bound for the A-changes that can affect $\Phi_d(A, D, x)$ while x is tracked by c.

Suppose then that a change in D is responsible for the destruction of a computation $\Phi_d(A, D, x)$. If A was correct up to the use of x, then $\hat{\Phi}_d(A, c)$ is correct. However, as a tracker, c has become useless: we cannot continue to use c as a tracker for future $\Phi_d(A, D, x)$ computations, as we are unable to redefine its use to match the new use of x. We must immediately cancel c, and replace it with a new tracker at the next expansionary stage for $\Phi_d(A, D)$.

Strategy Tree Construction:

A node τ working for requirement Q_d has outcomes ∞ and f, ordered $\infty <$ f, which measure $\limsup_s \operatorname{dom} \Phi_d(A, D)[s]$. The node $\tau\hat{}\infty$ has outcomes ∞_i, f_i for all $i < \omega$, ordered in the manner $\infty_0 < \mathrm{f}_0 < \infty_1 < \mathrm{f}_1 < \cdots$ which guess whether or not $\hat{\Phi}_\tau(A)$ is total. Each $\tau\hat{\ }\infty\hat{\ }\infty_i$ node has outcomes $j < \omega$, ordered $0 < 1 < 2 < \cdots$, where each node $\tau\hat{\ }\infty\hat{\ }\infty_i j$ guesses that $\hat{\Phi}_\tau(A)$ is the j^{th} member of the enumeration containing all total α-c.a. functions. The nodes $\tau\hat{\ }\mathrm{f}$, $\tau\hat{\ }\infty\hat{\ }\mathrm{f}_i$ for all i, and $\tau\hat{\ }\infty\hat{\ }\infty_i j$ for all i, j all work for the next, lower priority requirement.

The node τ is responsible for the enumeration of the shadow functional $\hat{\Phi}_\tau(A)$, such that if $\Phi_d(A, D)$ is total, then $\hat{\Phi}_\tau(A)$ is also total.

Since A is low$_2$, the set of indices of functionals Ψ such that $\Psi(A)$ is total is Σ^0_3. We can translate the question of membership in a Π^0_2 set into whether or not a given non-decreasing sequence is bounded. By the recursion theorem, the index of the functional $\hat{\Phi}_\tau(A)$ enumerated by τ is known to us. We thus obtain a computable list $l_s(\tau, n)$ of sequences, non-decreasing in s, such that $\hat{\Phi}_\tau(A)$ is total if and only if the sequence $\langle l_s(\tau, n)\rangle$ is unbounded for some n. It is this list of sequences that we check against when $\tau\hat{\ }\infty$ is accessible.

A node σ working for requirement P^i_e has outcomes ∞ and f, with $\infty <$ f, which guess whether σ will act infinitely or finitely often (respectively). Both of these outcomes work for the next (lower) priority requirement.

The node σ is responsible for the enumeration of the functional $\Delta_\sigma(A, D)$, to the end that either $\Delta_\sigma(A, D) = W_e$, or $\Lambda_e(D, W_e) \neq f^i$. (The functionals Λ_e are all enumerated at the root node.)

Suppose τ works for requirement Q_d; we let $\tau \in \mathrm{prec}_\infty(\sigma)$ if $\tau\hat{\ }\infty\hat{\ }\infty_n \prec \sigma$ for some n, and let $\tau \in \mathrm{prec}_{\mathrm{f}}(\sigma)$ if $\tau\hat{\ }\infty\hat{\ }\mathrm{f}_y \preceq \sigma$ for some y. Suppose a follower $p = p(\sigma, j)$ of σ is appointed at stage s. Let $t > s$; if t is least such that σ is accessible at stage t and $o^i_t(p)\downarrow$, for each $\tau \in \mathrm{prec}_\infty(\sigma)$ we define $m^\tau(p)$ and we declare p to be *realised* from stage t onward. Our intent for the value $m^\tau(p)$ is that, supposing τ works for Q_d, any computation $\Phi_d(A, D, x)$ for $x < m^\tau(p)$ is protected from the machinations of p; that is, at any stage s, we want $\lambda_{e,s}(p) > \varphi_{d,s}(x)$.

For $x < \omega$, *p may not injure* $\Phi_d(A, D, x)$ if τ works for Q_d, and either:

(i) $\tau \in \mathrm{prec}_\infty(\sigma)$, and $x < m^\tau(p)$; or,

(ii) $\tau \in \mathrm{prec}_{\mathrm{f}}(\sigma)$, and $x < y$.

We may also refer to such a $\Phi_d(A, D, x)$ as a *protected computation* (w.r.t. p). If p is realised and $j \downarrow W_{e,r}$, and we increase the use of p to be large at stage r, then we refer to p as *permitted* from stage r onward.

Let all requirements be ordered in order type ω, and let s be a stage.

First at s, we must check for any follower on the tree for which we may lift its use. We do this by asking: is there a node σ working for a requirement P_e^i with a realised follower $p = p(\sigma, j)$, such that $j \in W_{e,s} \setminus W_{e,s-1}$? If there is, then $\Lambda_e(D, W_e, p) \uparrow [s]$, as $j < \lambda_{e,s}(p) = p$; we take the strongest such σ, cancel all followers p' of σ such that $p' > p$, and redefine $\Lambda_e(D, W_e, p)[s+1] = s+1$ with large use. We now consider this p to be *permitted*, and initialise all nodes weaker than $\sigma\hat{}\mathrm{f}$, inclusive. We repeat this entire process until no further followers may be permitted, then proceed to define the collection γ_s of accessible nodes.

Let $\tau \in \gamma_s$ work for requirement Q_e. Let $t < s$ be the last stage before s at which $\tau\hat{}\infty$ was accessible, or $t = 0$ if there is no such stage. If $\mathrm{dom}\, \Phi_e(A, D)[s] \leqslant t$, let $\tau\hat{}\mathrm{f}$ be next accessible (added to γ_s); otherwise, s appears expansionary for $\Phi_e(A, D)[s]$, so we let $\tau\hat{}\infty \in \gamma_s$ instead.

Suppose $\tau\hat{}\infty \in \gamma_s$. Let y be least such that either $\Phi_e(A, D, y) \uparrow [t]$ or the computation $\Phi_e(A, D, y)[t]$ was destroyed since stage t. For each $n < s$, let t_n be the last stage before s at which $\tau\hat{}\infty\hat{}\infty_n$ was accessible, or $t_n = 0$ if there is no such stage. If there is some $n \leqslant y$ such that $l_s(\tau, n) \geqslant t_n$, then we guess that the sequence $l(\tau, n)$ is unbounded (i.e. that $\hat{\Phi}_\tau(A)$ is total), and we let $\tau\hat{}\infty\hat{}\infty_n$ be next accessible for the least such n. If there is no such $n \leqslant y$, we let $\tau\hat{}\infty\hat{}\mathrm{f}_y$ be next accessible.

At this point, when $\tau\hat{}\infty$ is accessible, we take the opportunity to maintain our shadow functional, $\hat{\Phi}$. Let $x < \omega$ be such that $c = \mathrm{tr}_s(\tau, x)$ is defined. If $\hat{\Phi}_\tau(A, c) \uparrow [s]$, and either

- $\tau\hat{\ }\infty\hat{\ }\mathrm{f}_y$ is next accessible, and $x < y$; or,
- $\tau\hat{\ }\infty\hat{\ }\infty_n$ is next accessible, and $x < t$

holds, we define $\hat{\Phi}_{\tau,s+1}(A_s, c) = s$ with use $\varphi_{e,s}(x)$. If any $c < s$ is not a tracker for any x at τ, and $\hat{\Phi}_\tau(A, c) \uparrow [s]$, we define $\hat{\Phi}_{\tau,s+1}(A_s, c) = 0$ with use 0. To conclude this series of (possible) definitions, for all $x < s$ that have no defined tracker, we define a new, large tracker $\mathrm{tr}_{s+1}(\tau, x)$.

Suppose then that $\tau\hat{\ }\infty\hat{\ }\infty_n$ is accessible at s for some n. For each $i < s$, let r_i be the last stage before s at which $\tau\hat{\ }\infty\hat{\ }\infty_n i$ was accessible, or $r_i = 0$ if there was no such stage. Recall that the sequence $\langle g^i \rangle_{i<\omega}$ contains every α-c.a. function. We let $\tau\hat{\ }\infty\hat{\ }\infty_n i$ be next accessible for the least $i \leqslant s$ such that for all $x < r_i$, $c = \mathrm{tr}_s(\tau, x)$ is defined, $m^i_s(c) \downarrow$, and $\hat{\Phi}_\tau(A, c)[s] = g^i_s(c)$. We note that $r_s = 0$, so such an i must exist, even if it merely satisfies these conditions trivially.

Let $\sigma \in \gamma_s$ work for requirement P^i_e. There are several possible circumstances for σ which require individual attention.

(i) σ has permitted follower $p = p(\sigma, j)$.

(a) If $\Delta_\sigma(A, D, j) \uparrow [s]$, cancel the follower p, and set $\Delta_\sigma(A, D, j) = 1$ with use 0. We then let $\sigma\hat{\ }\infty$ be next accessible.

(b) If $\Delta_\sigma(A, D, j) \downarrow [s]$ and $\Lambda_e(D, W_e, p)[s] \neq f^i_s(p)$, we let $\sigma\hat{\ }\mathrm{f}$ be next accessible.

(c) If $\Delta_\sigma(A, D, j) \downarrow [s]$ and $\Lambda_e(D, W_e, p)[s] = f^i_s(p)$, we enumerate $\lambda_{e,s}(p)$ into D_{s+1}, and redefine $\Lambda_e(D, W_e, p)[s+1] = s+1$ with large use. We then initialise all nodes weaker than $\sigma\hat{\ }\mathrm{f}$ (inclusive), and cancel all trackers for $x \geqslant m^\tau(p)$ for all $\tau \in \mathrm{prec}_\infty(\sigma)$. We then maintain the functional Λ as detailed at the end of the construction, and complete the stage s.

(ii) σ has a follower p such that $o^i_s(p)\uparrow$. We immediately let $\sigma\hat{}\mathrm{f}$ be next accessible.

(iii) For every follower p of σ, $o^i_s(p)\downarrow$ but p is not permitted; this includes if σ has no followers, in which case we skip straight to appointing a new follower.

Let $t < s$ be the last stage at which σ was accessible. If there is a follower $p = p(\sigma, j)$ of σ for which either: (a) $o^i_t(p)\uparrow$ while $o^i_s(p)\downarrow$, or (b) $\Delta_\sigma(A, D, j)\uparrow[s]$, we proceed to define $\Delta_\sigma(A, D, j)[s+1] = W_{e,s+1}(j)$. If $W_{e,s}(j) = 1$, we set the use $\delta_{\sigma,s+1}(j)$ to be 0 and cancel p. If $W_{e,s}(j) = 0$, then we set the use to be:

$$\delta_{\sigma,s+1}(j) = \max\{\varphi_{d,s}(x) : p \text{ may not injure } \Phi_d(A, D, x)\} + 1.$$

Finally, if (a) holds and if $W_{e,s}(j) = 0$, then this is the stage at which p becomes realised. Then for each $\tau \in \mathrm{prec}_\infty(\sigma)$, if τ works for Q_d we define $m^\tau(p) = \mathrm{dom}\Phi_d(A, D)[s]$.

Let k be largest such that $p(\sigma, k)$ is a follower of σ. Let $m > k$ be least such that $m \notin W_{e,s}$. For any $k < n < m$, if $\Delta_\sigma(A, D, n)\uparrow[s]$, define $\Delta_\sigma(A, D, n)[s+1] = 1$ with use 0. We then appoint a new, large follower $p(\sigma, m)$, define $\Lambda_e(D, W_e, p)[s+1] = s+1$ with use $\lambda_{e,s+1}(p) = p$, and let $\sigma\hat{}\infty$ be next accessible.

We end stage s when $|\gamma_s| = s$, if not ended sooner by enumeration into D. At the conclusion of stage s, we maintain the functional Λ to ensure its totality. For any pair $(e, p) \leqslant s$ for which $\lambda_e(p)$ was not already redefined at s, if $p \notin \mathrm{dom}\Lambda_{e,s}(D_{s+1}, W_{e,s+1})$ we let $\Lambda_e(D, W_e, p)[s+1] = s+1$. If p is a follower of a node σ working for requirement P^i_e, let $\lambda_{e,s+1}(p) = \lambda_{e,s}(p)$;

otherwise, let $\lambda_{e,s+1}(p) = 0$.

Verification:

For a node μ on the tree, we denote s to be a *μ-stage* if $\mu \in \gamma_s$. First, we prove that a follower p will never injure a computation that is protected from it.

LEMMA 2.1.4: *Let σ be a node working for requirement P^i_d, and let $p = p(\sigma, j)$ be a follower. Suppose that p is permitted at or prior to stage s, where s is a σ-stage.*
Let $r < s$ be the last stage at which $\Delta_\sigma(A, D, j)$ was defined. If p is not cancelled at s, then for every computation $\Phi_e(A, D, x)$ that p may not injure, we have:

$$\Phi_e(A, D, x)[r] \downarrow = \Phi_e(A, D, x)[s].$$

Proof. Since $\Delta_\sigma(A, D, j)$ is defined at r, r is a σ-stage. Suppose $\Phi_e(A, D, x)$ may not be injured by p, then there is a node τ working for Q_e such that either:

(i) $\tau \in \mathrm{prec}_\infty(\sigma)$, and $x < m^\tau(p)$; or,

(ii) $\tau\hat{\ }\infty\hat{\ }\mathrm{f}_y \preceq \sigma$, and $x < y$.

If (i) is the case, then r is a $\tau\hat{\ }\infty\hat{\ }\infty_n$ stage for some n. Hence, $\mathrm{dom}\Phi_e(A, D)[r] \geqslant m^\tau(p)$, so certainly $\Phi_e(A, D, x) \downarrow [r]$. If instead (ii) holds, then y is least such that $\Phi_e(A, D, y)$ was destroyed at or since the previous $\tau\hat{\ }\infty$-stage. Hence, for all $x < y$, we have $\Phi_e(A, D, x) \downarrow [r]$. Thus, for all protected computations $\Phi_e(A, D, x)$, at r we define $\delta_{\sigma,r}(j) > \varphi_{e,r}(x)$.
Suppose p is not cancelled at s, then $\Delta_\sigma(A, D, j)[s] \downarrow = \Delta_\sigma(A, D, j)[r]$, and $\delta_{\sigma,s}(j) = \delta_{\sigma,r}(j)$. Hence, for any protected computation $\Phi_e(A, D, x)$, we have $\delta_{\sigma,s}(j) > \varphi_{e,r}(x)$. Since $\Delta_\sigma(A, D, j)[r]$ persists until stage s, there is no change in either A or D below $\delta_{\sigma,s}(j)$; there must then be no change below

$\varphi_{e,r}(x)$ on the interval $[r, s)$. Therefore, $\Phi_e(A, D, x)[r] = \Phi_e(A, D, x)[s]$.

□

COROLLARY 2.1.5: *Let $e, x < \omega$ be fixed. Let σ working for P^i_d have a follower $p = p(\sigma, j)$; suppose σ enumerates $\lambda_{d,s}(p)$ into D_{s+1}. If $\Phi_e(A, D, x)$ is a protected computation with respect to p, then $\Phi_e(A, D, x) \downarrow [s]$ and $\lambda_{d,s}(p) > \varphi_{e,s}(x)$.*

Proof. Since σ enumerates $\lambda_{d,s}(p)$ into D_{s+1}, the follower p must have been permitted prior to stage s. Let $t < s$ be this stage. Then the use of p was lifted at this stage to be larger than all computations protected from p. If any such computation did not converge at stage t, then by Lemma 2.1.4, p would be cancelled at or before stage s. Hence, for all protected computations $\Phi_e(A, D, x)$, when $\lambda_{d,t}(p)$ is lifted to be large we have $\lambda_{d,t}(p) > \varphi_{e,t}(x)$.
If for any protected computation $\Phi_e(A, D, x)$ we have $\varphi_{e,s}(x) \neq \varphi_{e,t}(x)$ (or indeed, if $\Phi_e(A, D, x) \uparrow [s]$), then $\Phi_e(A, D, x)[s] \neq \Phi_e(A, D, x)[t]$ and by Lemma 2.1.4 p is cancelled at, or prior to, stage s. Hence, $\Phi_e(A, D, x) \downarrow [s]$ and $\varphi_{e,s}(x) = \varphi_{e,t}(x)$, so $\lambda_{d,s}(p) > \varphi_{e,s}(x)$.

□

Hence, if p is permitted at a stage at which $\Phi_e(A, D, x) \uparrow$, where $\Phi_e(A, D, x)$ is protected from p, p will be cancelled at the next σ-stage thereafter and thus cannot injure the computation. Further, while p is not cancelled, its use is always too large to injure $\Phi_e(A, D, x)$.
We now need to be sure that, if a computation is destroyed by change in D, its tracker is cancelled immediately to allow us to correctly anticipate A-change.

LEMMA 2.1.6: *Let τ be a node working for requirement Q_e. Let s be a stage; let $x < \omega$ be such that $c = tr_s(\tau, x)$ is defined. Suppose that $\hat{\Phi}_\tau(A, c) \downarrow [s]$, and let $u = \hat{\varphi}_{\tau,s}(c)$. Then:*

(i) $\Phi_e(A, D, x) \downarrow [s]$ *and* $u = \varphi_{e,s}(x)$*; and,*

(ii) If $D_s \upharpoonright u \neq D_{s+1} \upharpoonright u$, then the tracker c is cancelled at s.

Proof. Suppose (i) and (ii) hold up to stage s, and that the lemma hypotheses hold at s. Let $t < s$ be the stage at which the computation $\hat{\Phi}_\tau(A, c)[s]$ was defined; then $A_t \upharpoonright u = A_s \upharpoonright u$. Applying the lemma at stage t, we have $\Phi_e(A, D, x) \downarrow [t]$ and $u = \varphi_{e,t}(x)$. Since trackers are chosen to be large, $c = \mathrm{tr}_t(\tau, x)$.
Since c is not cancelled in the interval $[t, s)$, using part (ii) we may infer that $D_t \upharpoonright u = D_{t+1} \upharpoonright u = \cdots = D_s \upharpoonright u$. Combined with the fact that $A_t \upharpoonright u = A_s \upharpoonright u$, this show that the computation $\Phi_e(A, D, x)[t]$ is not destroyed in the interval $[t, s)$; hence, (i) holds at s.

Suppose that $D_s \upharpoonright u \neq D_{s+1} \upharpoonright u$; then at s we enumerate a number $\lambda_{d,s}(p) < u$ into D, where $p = p(\sigma, j)$ and σ works for requirement P^i_d. By Corollary 2.1.5, p is allowed to injure the computation $\Phi_e(A, D, x)[s]$, and since followers are appointed large, we may infer that p was appointed, realised and permitted prior to stage t. If τ is weaker than $\sigma\hat{}\mathrm{f}$ (inclusive), then τ is initialised at s and c is thus cancelled. If $\tau \succeq \sigma\hat{}\infty$, we note that by construction we only let $\sigma\hat{}\infty$ be accessible at s if $\Delta_\sigma(A, D, j)[s] \uparrow$. Then we cancel p at s upon visiting σ, so σ cannot enumerate $\lambda_{d,s}(p)$ into D at this stage.
Let r be the last $\tau\hat{}\infty$-stage before s. If $\sigma \succeq \tau\hat{}\mathrm{f}$, or σ is to the right of τ, then σ was initialised at r. Since $r \geqslant t$ and p was permitted by stage t, p was cancelled by this initialisation.
If $\sigma \succeq \tau\hat{}\infty\hat{}\mathrm{f}_y$, then since $\Phi_e(A, D, x) \downarrow [s]$ we may infer that $x < y$. Then $\Phi_e(A, D, x)$ is protected from p, so enumeration by σ for p would not alter $D_s \upharpoonright u$.
Finally, if $\sigma \succeq \tau\hat{}\infty\hat{}\infty_n$ we must have $x \geqslant m^\tau(p)$, as $\Phi_e(A, D, x)$ is not protected from p. Then, when σ enumerates $\lambda_{d,s}(p)$ into D_{s+1} at s, we cancel c by construction.

□

Let γ_ω be the true path, the leftmost path of the tree visited infinitely often.

LEMMA 2.1.7: *Suppose $\tau \in \gamma_\omega$ works for Q_e. Then $\Phi_e(A, D)$ is total if and only if for some $n < \omega$, $\tau\hat{}\infty\hat{}\infty_n$ is on the true path if and only if $\hat{\Phi}_\tau(A)$ is total.*

Proof. Suppose that $\Phi_e(A, D)$ is total so $\tau\hat{}\infty$ is on the true path, and let $c < \omega$. If c is chosen as a tracker for some x and later cancelled, let t be the stage at which c is cancelled; if c is never chosen as a tracker, let $t = c$. Let s be the least stage $s > t$ at which $\tau\hat{}\infty$ is accessible and $\hat{\Phi}_\tau(A, c) \uparrow [s]$. We then define $\hat{\Phi}_{\tau,s+1}(A_s, c) = 0$ with use 0.
Suppose then that c is chosen as a tracker for some x at stage r and is never cancelled. Since $\Phi_e(A, D)$ is total, there must be a stage s beyond which all $y \leqslant x$ are in dom $\Phi_e(A, D, y)[t]$ for all $t \geqslant s$. Therefore, it must be that eventually no f_y outcome is ever guessed for $y \leqslant x$. At every stage t beyond s at which $\tau\hat{}\infty$ is accessible, if $\hat{\Phi}(A, c) \uparrow [t]$, then we redefine $\hat{\Phi}_{\tau,t+1}(A_t, c) = t$ with use $\varphi_{e,t}(x)$. This use must stabilise, and eventually A must stabilise below that use, so there will eventually be an A-correct computation $\hat{\Phi}_\tau(A, c)$. Therefore, $\hat{\Phi}_\tau(A)$ is total, and so there must then be an n such that $\langle l_s(\tau, n)\rangle$ is an unbounded sequence; for the least such n, $\tau\hat{}\infty\hat{}\infty_n$ must be on the true path.

Suppose instead that $\limsup_s \operatorname{dom} \Phi_e(A, D)[s] < \infty$. Then there is a stage s such that for all $t > s$, we have $\tau\hat{}\infty \notin \gamma_t$. Since we only define $\hat{\Phi}_\tau(A)$ at $\tau\hat{}\infty$ stages, $\hat{\Phi}_\tau(A)$ is necessarily partial.
Suppose instead that $\tau\hat{}\infty$ is accessible infinitely often. Let $x < \omega$. Any tracker $\mathrm{tr}_s(\tau, x)$ can only be cancelled by a node $\sigma \succ \tau\hat{}\infty$ working for a requirement P_d^i, with a follower $p = p(\sigma, j)$, such that $x \geqslant m^\tau(p)$. Let $p' = p(\mu, k)$ be a follower of a node working for a P requirement such that $\mu \succ \tau\hat{}\infty$. The value $m^\tau(p)$ is static and defined at the σ-stage at which p becomes realised; hence if $m^\tau(p)$ is defined at stage s and $m^\tau(p')$ defined at stage t, with $s < t$, then $m^\tau(p) < m^\tau(p')$. This is because s, t are both $\tau\hat{}\infty$-stages; $m^\tau(p) = \operatorname{dom} \Phi_e(A, D)[s] < s$ and $m^\tau(p') = \operatorname{dom} \Phi_e(A, D)[t] \geqslant r \geqslant s$, where r is the last $\tau\hat{}\infty$-stage prior to t.

Therefore, for any x, the number of pairs $\langle \sigma, p \rangle$ such that $x \geqslant m^\tau(p)$ must be finite. Each $\langle \sigma, p \rangle$ pair may only cause finitely many cancellations of a tracker of x. Otherwise σ acts infinitely often for p; let $\{s_j : j \in \omega\}$, with $s_0 < s_1 < \cdots$, be the stages at which this occurs. If σ acts at a stage s, it must be because $\Lambda_e(D, W_e, p)[s] = f^i_s(p)$. We then redefine $\Lambda_e(D, W_e, p)[s+1] = s+1 \neq f^i_s(p)$. Then $f^i_{s_0}(p) \neq f^i_{s_1}(p) \neq \cdots$, and hence we have an infinite descending chain of ordinals $o^i_{s_0}(p) > o^i_{s_1}(p) > \cdots$ - a contradiction.
Hence x can have its tracker cancelled only finitely many times, so there must eventually be a tracker for x which is never cancelled.

Let $y = \mathrm{dom}\Phi_e(A, D)$, and let c be the tracker that is eventually assigned to y and never cancelled. If $\hat{\Phi}_\tau(A, c) \downarrow$ with use u, then eventually we will have a stage s at which both $A_s \upharpoonright u$ and $D_s \upharpoonright u$ are correct. By Lemma 2.1.6, we would then have a computation for $\Phi_e(A, D, y)$ correct in both A and D - a contradiction. Hence if $\Phi_e(A, D)$ is partial, then $\hat{\Phi}_\tau(A)$ is also partial.

□

LEMMA 2.1.8: *The true path, γ_ω, is infinite. The construction is then fair to nodes on the true path.*

Proof. Suppose μ is a node on the true path; we must show that a child of μ is also on the true path. If μ does not work for a P requirement, then $\mu = \tau$, $\mu = \tau\hat{}\infty$, or $\mu = \tau\hat{}\infty\hat{}\infty_n$ for some n, where τ works for some requirement Q_e. By Lemma 2.1.7, if $\mu = \tau\hat{}\infty$, then there is an n such that $\tau\hat{}\infty\hat{}\infty_n$ is on the true path. If $\mu = \tau\hat{}\infty\hat{}\infty_n$ for some n and no one outcome is accessible infinitely often, then $\hat{\Phi}_\tau(A) \neq f^i$ for any f^i in the enumeration containing all α-c.a. functions; this contradicts the fact that A is totally α-c.a..
Otherwise, μ has finitely many outcomes. In any case, (at least) one of the child nodes of μ must be accessible infinitely often, and therefore on the true path.

Otherwise, μ works for a requirement P_e^i. For neither child of μ to be accessible infinitely often, μ must act and end the stage at almost every stage that it is accessible.

As shown in the proof of Lemma 2.1.7, μ cannot act infinitely often on any particular follower. Then μ must have infinitely many followers permitted and later cancelled. However, when we cancel a follower of μ, we let $\mu\hat{}\infty$ be next accessible. Hence, the child node $\mu\hat{}\infty$ is accessible infinitely often, and thus on the true path.

$\square$

The true path then contains, for every P and Q requirement, a node that works for it. If we can prove that every node which appears on the true path is successful in meeting its requirement, then Theorem 2.1.3 is proved. Firstly, we must ensure that for every $e < \omega$, the functional Λ_e is total; otherwise, even if $\Lambda_e(D, W_e, p) \neq f^i(p)$ for some follower p of a node σ working for P_e^i, we cannot support a claim that $\deg_{\mathrm{T}}(A \oplus D \oplus W_e)$ is not totally β-c.a..

Lemma 2.1.9: *For all $e < \omega$, $\Lambda_e(D, W_e)$ is total.*

Proof. Fix e, and let $p < \omega$. Let $t = 0$ if p is never appointed as a follower, or, if p is appointed to a node working for P_e^i and later cancelled, let t be the stage of cancellation. Then at the least stage $s \geqslant t$ such that $s \geqslant (e, p)$, at the conclusion of the stage we define $\Lambda_e(D, W_e, p)[s+1] = s+1$ with use 0.

Suppose then that p is appointed as a follower to a node σ working for P_e^i. Assuming p is not cancelled and is permitted, we redefine $\Lambda_e(D, W_e, p)$ at any stage thereafter where we see $\Lambda_e(D, W_e, p)[s] = f_s^i(p)$; we do so by letting $\Lambda_e(D, W_e, p)[s+1] = s+1 \neq f_s^i(p)$. We know that this can happen only finitely-many times, as shown in Lemma 2.1.7. Let t be the stage at which p is permitted; there is then a final stage $s^* \geqslant t$ such that, for all $s \geqslant s^*$, $\Lambda_e(D, W_e, p)[s] \neq f_s^i(p)$.

After this stage, we only redefine $\Lambda_e(D, W_e, p)$ when it is destroyed by enumeration into A or D below the use $\lambda_{e,s^*}(p)$. If p is never permitted, this is the only occasion at which we redefine the computation. In this case, let s^* be the stage at which p was appointed. Then, when we redefine $\Lambda_e(D, W_e, p)$, the use remains the same; hence, if $\Lambda_e(D, W_e, p)$ is redefined infinitely often, then either A, D, or W_e change infinitely often below $\lambda_{e,s^*}(p)$. This cannot be the case, so there is a stage $s \geqslant s^*$ at which $\Lambda_e(D, W_e, p)[s] = \Lambda_e(D, W_e, p)$.

□

LEMMA 2.1.10: *For all e and all i, the requirement P_e^i is met.*

Proof. Let σ be a node on the true path working for requirement P_e^i. There are three possible circumstances for σ.

(i) σ has a follower p that is never realised;

(ii) σ (eventually) has a follower p that is realised, permitted, and never cancelled;

(iii) every follower appointed to σ that is permitted is eventually cancelled.

In case (i), we have $o^i(p)\uparrow$, and P_e^i is met by false hypothesis.

Suppose then that case (ii) holds, and there is a follower $p = p(\sigma, j)$ of σ such that p is permitted at some stage s' and never cancelled thereafter. We know σ must only act finitely often for p, so let $t \geqslant s'$ be the least stage such that σ never acts for p after stage t.
Suppose that, despite our action for p at σ, $f^i(p) = \Lambda_e(D, W_e, p)[t + 1]$. At stage t we defined $\Lambda_e(D, W_e, p)[t+1] = t+1 \neq f_t^i(p)$. Therefore $f^i(p) \neq f_t^i(p)$, so there is a stage $s > t$ at which we see $f_s^i(p) = f^i(p)$. Since σ is on the true path, it must become accessible again at some stage $r \geqslant s$; at this stage, we redefine $\Lambda_e(D, W_e, p)[r] \neq f^i(p)$.
Hence, case (ii) meets the requirement P_e^i.

Finally, suppose that case (iii) holds, and let $j < \omega$. Then $\sigma\hat{}\infty$ is accessible infinitely often, and hence is on the true path. In this case, we require that

$\Delta_\sigma(A, D) = W_e$.
Let s^* be the final stage at which σ is initialised, and let $p = p(\sigma, j)$ be the follower appointed to σ for j after s^*. If no such p exists, there must be a stage $s > s^*$ at which we consider appointing a follower but choose not to because we see $j \in W_{e,s}$ already. Then we define $\Delta_\sigma(A, D, j)[s + 1] = 1 = W_e(j)$ with use 0, which is permanent. If the follower p exists but is cancelled at some stage $s > s^*$, it is cancelled because of A-change which occurred after p was permitted, i.e. after $j \downarrow W_e$. At the next σ-stage we set $\Delta_\sigma(A, D, j)[s + 1] = 1 = W_e(j)$ with use 0.
Suppose then that $p = p(\sigma, j)$ exists, and is not cancelled; p must not become permitted or it is in fact case (ii), so $W_e(j) = 0$. Suppose that $\lim_s \delta_{\sigma,s}(j) = \infty$. Then $\delta_\sigma(j)$ must be redefined infinitely often, and thus there are infinitely many stages s at which $\Delta_\sigma(A, D, j) \uparrow [s]$. At each such s where σ is accessible, we redefine $\Delta_\sigma(A, D, j)[s + 1] = W_{e,s+1}(j) = 0$ with use $\delta_{\sigma,s+1}(j) = \max\{\varphi_{d,s}(x) : x < m^\tau(p),\ \tau \in \mathrm{prec}_\infty(\sigma),\ \tau \text{ works for } Q_d\} + 1$. The set $\{x : x < m^\tau(p)\}$ for any $\tau \in \mathrm{prec}_\infty(\sigma)$ is fixed at the σ-stage at or immediately following the first stage t at which $o^i_t(p) \downarrow$, and is never altered. Hence if $\lim_s \delta_{\sigma,s}(j) = \infty$, then there is a $\tau \in \mathrm{prec}_\infty(\sigma)$ such that $\lim_s \varphi_{d,s}(x) = \infty$ for some $x < m^\tau(p)$. However, $\tau \in \mathrm{prec}_\infty(\sigma)$, so $\tau\hat{\ }\infty\hat{\ }\infty_n$ is on the true path for some n. By Lemma 2.1.7, $\Phi_d(A, D)$ must be total, so $\lim_s \varphi_{d,s}(y)$ is finite for every $y < \omega$ - a contradiction. Hence, $\lim_s \delta_{\sigma,s}(j)$ is finite, so $\Delta_\sigma(A, D, j) \downarrow = 0 = W_e(j)$.

$\square$

To show that the Q requirements are met, we require the commutative addition operation on ordinals. Let γ, δ be ordinals with respective Cantor normal forms $\gamma = \omega^{\gamma_1} \cdot n_1 + \cdots + \omega^{\gamma_k} \cdot n_k$, $\delta = \omega^{\delta_1} \cdot m_1 + \cdots + \omega^{\delta_l} \cdot m_l$. Let S be the set formed by collecting all exponents that appear in each form, i.e. let $S = \{\gamma_i \mid 1 \leqslant i \leqslant k\} \cup \{\delta_i \mid 1 \leqslant i \leqslant l\}$, and let the members of S be ordered $\alpha_1 > \alpha_2 > \cdots > \alpha_n$ by the usual ordering on ordinals.
We then write $\gamma = \omega^{\alpha_1} \cdot q_1 + \cdots + \omega^{\alpha_n} \cdot q_n$, where for all $1 \leqslant i \leqslant n$, we let $q_i = n_j$ if $\alpha_i = \gamma_j$ and let $q_i = 0$ otherwise. Similarly, $\delta = \omega^{\alpha_1} \cdot r_1 + \cdots + \omega^{\alpha_n} \cdot r_n$,

where for all $1 \leqslant i \leqslant n$, $r_i = m_j$ if $\alpha_i = \delta_j$ and $r_i = 0$ otherwise.
The *commutative sum* of γ and δ is then:

$$\gamma \oplus \delta = \omega^{\alpha_1} \cdot (q_1 + r_1) + \cdots + \omega^{\alpha_n} \cdot (q_n + r_n).$$

For ordinals α, β, γ, we then have $\gamma \oplus \beta = \beta \oplus \gamma$ and $(\alpha+\beta)+\gamma = \alpha+(\beta+\gamma)$, by virtue of the commutativity and associativity of addition on the natural numbers. We borrow directly from [4] the following lemmata of use.

LEMMA 2.1.11: *Any power of ω is closed under $\oplus$.*

LEMMA 2.1.12: *Let $\beta_1, \beta_2, \ldots, \beta_n$ and $\gamma_1, \gamma_2, \ldots, \gamma_n$ be two n-tuples of ordinals. Suppose that for all $i \leqslant n$, $\beta_i \leqslant \gamma_i$. Then $\bigoplus_{i \leqslant n} \beta_i \leqslant \bigoplus_{i \leqslant n} \gamma_i$, and $\bigoplus_{i \leqslant n} \beta_i < \bigoplus_{i \leqslant n} \gamma_i$ if, and only if, there is some $i \leqslant n$ such that $\beta_i < \gamma_i$.*

We now prove that the Q requirements are met.

LEMMA 2.1.13: *For all $e < \omega$, the requirement Q_e is met.*

Proof. Suppose that $\Phi_e(A, D)$ is total, and τ is the node on the true path working for Q_e. We proceed to build functions $\hat{f}, \hat{o}$ such that $\langle \hat{f}_s, \hat{o}_s \rangle_{s<\omega}$ is a β-computable approximation of $\Phi_e(A, D)$.
Since $\Phi_e(A, D)$ is total, by Lemma 2.1.7 there are $n, i < \omega$ such that $\rho = \tau\hat{}\infty\hat{}\infty_n\hat{}i$ is on the true path. Let s^* be the last stage at which ρ is initialised, and let $s_0 < s_1 < s_2 < \cdots$ be the stages after s^* at which ρ is accessible.

Fix $x < \omega$. Let $i(x)$ be the least i such that $x < \mathrm{dom}\Phi_e(A, D)[s_{i-1}]$. Let $a(x)$ be the collection of pairs (σ, p) such that $\tau \in \mathrm{prec}_\infty(\sigma)$ or $\tau\hat{}\infty\hat{}\mathrm{f}_y \preceq \sigma$ for $y > x$, and p is a follower for σ that was realised before stage $s_{i(x)}$ but not cancelled by stage $s_{i(x)}$. Then $a(x)$ comprises all followers that may injure the computation $\Phi_e(A, D, x)$ by enumerating into D.

For all $x < \omega$ and all $j \geqslant i(x)$, we define $\hat{f}_j(x) = \Phi_e(A, D, x)[s_j]$.

For all $j \geqslant i(x)$, let $a_j(x)$ be the collection of pairs $(\sigma, p) \in a(x)$ such that σ is not initialised at any stage $r \in [s_{i(x)}, s_j)$, and p is still a follower for σ at the beginning of stage s_j. We note that for any $j \geqslant i(x)$, we have $\Phi_e(A, D, x) \downarrow [s_j]$, $c_j = c_j(x) = \mathrm{tr}_{s_j}(\tau, x)$ is defined, $m^i_{s_j}(c_j) \downarrow$, and $\hat{\Phi}_\tau(A, c_j) \downarrow [s_j] = g^i_{s_j}(c_j)$.

Suppose $\langle \sigma_1, p_1 \rangle, \ldots, \langle \sigma_n, p_n \rangle$ are the members of $a_j(x)$, and that σ_k works for the requirement $P^{i_k}_{d_k}$. For each k, let $t_{k,j}$ be the last stage prior to s_j at which σ_k acted for p_k (including permission), or if there is no such stage, let $t_{k,j}$ be the stage at which p_k was appointed.
Let $\eta_j(x) = o^{i_1}_{t_{1,j}}(p_1) \oplus o^{i_2}_{t_{2,j}}(p_2) \oplus \cdots \oplus o^{i_n}_{t_{n,j}}(p_n)$.
For $j \geqslant i(x)$ we then define:

$$\hat{o}_j(x) = \alpha \cdot \eta_j(x) + m^i_{s_j}(c).$$

Let $j \geqslant i(x)$, and let $u = \varphi_{e,s_j}(x)$. If $\hat{f}_j(x) \neq \hat{f}_{j+1}(x)$, and this discrepancy is caused by change in A alone, we see $D_{s_j} \upharpoonright u = D_{s_{j+1}} \upharpoonright u$, but $A_{s_j} \upharpoonright u \neq A_{s_{j+1}} \upharpoonright u$. By Lemma 2.1.6, $u = \hat{\varphi}_\tau(c_j)[s_j]$, so the enumeration into A that destroys the computation $\Phi_{e,s_j}(A, D, x)$ will also destroy the computation $\hat{\Phi}_\tau(A, c_j)[s_j]$. When $\hat{\Phi}_\tau(A, c_j)$ is redefined, it is defined to be the stage number, so certainly $\hat{\Phi}_\tau(A, c_j)[s_{j+1}] > s_j$. In particular, $\hat{\Phi}_\tau(A, c_j)[s_j] \neq \hat{\Phi}_\tau(A, c_j)[s_{j+1}]$. Then $g^i_{s_j}(c_j) \neq g^i_{s_{j+1}}(c_j)$, and it follows that $m^i_{s_j}(c_j) > m^i_{s_{j+1}}(c_j)$.

If $\hat{f}_j(x) \neq \hat{f}_{j+1}(x)$ is caused by change in D, then $D_{s_j} \upharpoonright u \neq D_{s_{j+1}} \upharpoonright u$. This change in D is caused by a pair $\langle \sigma, p \rangle = \langle \sigma_k, p_k \rangle \in a_j(x)$, by σ enumerating $\lambda_{d_k, s_j}(p)$ into D at stage s_j in response to seeing $f^{i_k}_{s_j}(p) = \Lambda_{d_k}(D, W_{d_k}, p)[s_j]$. At stage $t_{k,j}$, we defined $\Lambda_{d_k}(D, W_{d_k}, p)[t_{k,j} + 1] = t_{k,j} + 1 > f^{i_k}_{t_{k,j}}(p)$. Hence $f^{i_k}_{s_j}(p) = \Lambda_{d_k}(D, W_{d_k}, p)[s_j]$ implies $f^{i_k}_{t_{k,j}}(p) \neq f^{i_k}_{s_j}(p)$, and consequently, given that $t_{k,j+1} = s_j$, we have $o^{i_k}_{t_{k,j}}(p) > o^{i_k}_{t_{k,j+1}}(p)$. By Lemma 2.1.12, $\eta_j(x) > \eta_{j+1}(x)$; then $\alpha \cdot \eta_j(x) \geqslant \alpha \cdot (\eta_{j+1}(x) + 1) > \alpha \cdot \eta_{j+1}(x) + m^i_{s_{j+1}}(c_{j+1})$.

Hence, whether the computation is destroyed by change in A or D, if $\hat{f}_j(x) \neq \hat{f}_{j+1}(x)$ then $\alpha \cdot \eta_j(x) + m^i_{s_j}(c_j) > \alpha \cdot \eta_{j+1}(x) + m^i_{s_{j+1}}(c_{j+1})$.

By definition, it is clear that $\lim_s \hat{f}_s = \Phi_e(A, D)$.
If $m^i_{s_j}(c_j) < m^i_{s_{j+1}}(c_{j+1})$, then $c_j \neq c_{j+1}$; this implies that $D_{s_j} \upharpoonright u \neq D_{s_{j+1}} \upharpoonright u$ for $u = \varphi_{e,s_j}(x)$, and as shown above, $\hat{o}_j(x) > \hat{o}_{j+1}(x)$ in this case. Otherwise, the function $\hat{o}$ inherits the non-increasing nature of the o^{i_k}, m^i functions that comprise it, and $\hat{o}$ is therefore non-increasing.
Since each $o^{i_k}(p_k)$ term is bounded by β and β is a power of ω, by Lemma 2.1.11 the sum $\eta_j(x)$ is also bounded by β. Since $\alpha \cdot \beta = \beta$, the term $\alpha \cdot \eta_j(x)$ is bounded by β. Finally, as β is closed under (regular) addition, $\hat{o}_j(x) = \alpha \cdot \eta_j(x) + m^i_{s_j}(c_j)$ is bounded by β.

Therefore, $\langle \hat{f}_s, \hat{o}_s \rangle_{s<\omega}$ is a β-computable approximation of $\Phi_e(A, D)$.

□

Every node on the true path then ensures that its requirement is met; this concludes the proof of Theorem 2.1.3.

2.2. A Construction Without Permitting

Our next theorem has a similar construction to Theorem 2.1.3, and borrows much of its verification. However, as we are not building a maximal totally ω^3-c.a. degree, we lack a mechanism by which to permit followers and lift their use. This alters which followers we can allow to injure a computation and prevents us from producing a bound lower than ω^3 in this construction. Further work might be to show that any totally ω-c.a. c.e. degree is bounded by a totally ω^2-c.a. c.e. degree which is not totally ω-c.a. (or by a maximal totally ω^2-c.a. c.e. degree). This has already been proven when $\boldsymbol{a}$ is a superlow c.e. degree, see Theorem 2.3.1.

THEOREM 2.2.1: *Let $\boldsymbol{a}$ be any totally ω-c.a., c.e. degree. There is a c.e. degree $\boldsymbol{b} > \boldsymbol{a}$ such that $\boldsymbol{b}$ is totally ω^3-c.a., but not totally ω-c.a..*

Fix $\boldsymbol{a}$, and let $\langle A_s \rangle_{s<\omega}$ be a given computable enumeration of $A \in \boldsymbol{a}$. We proceed to build a set D such that $\deg_{\mathrm{T}}(A \oplus D)$ is the required $\boldsymbol{b}$.

Requirements:

Let $\langle \Phi_e \rangle_{e<\omega}$ be an enumeration of all consistent functionals. Our requirements are simple; firstly, we require that $A \oplus D$ is totally ω^3-c.a.. This is achieved with the set of requirements

$$\text{for all } e < \omega,\ Q_e : \text{If } \Phi_e(A, D) \text{ is total, then it is } \omega^3\text{-c.a..}$$

We then, of course, require that $A \oplus D$ is not totally ω-c.a.. Let $\langle \langle f^i_s, o^i_s \rangle_{s<\omega} \rangle_{i<\omega}$ be an effective list of partial ω-computable approximations such that letting $f^i = \lim_s f^i_s$, the sequence $\langle f^i \rangle_{i<\omega}$ contains every ω-c.a. function. We enumerate a functional Λ, and diagonalise this against all members of $\langle f^i \rangle_{i<\omega}$. This process is captured in the requirements

$$\text{for all } i < \omega,\ P^i : \text{If } \textstyle\bigcup_s \operatorname{dom} o^i_s = \omega, \text{ then } \Lambda(A, D) \neq f^i.$$

Discussion:

To meet the P and Q requirements, we build a strategy tree. For a node σ to meet a requirement P^i in isolation, we appoint to it a follower p. Thereafter, whenever we see $\Lambda(A, D, p)[s] = f^i_s(p)$, we enumerate $\lambda_s(p)$ into D_{s+1} and redefine $\Lambda(A, D, p)[s + 1] = s + 1$ to be larger than all previously seen values of $f^i(p)$. A node τ working for Q_e will attempt to meet its requirement by appointing, to every $x < \omega$, an ordinal below ω^3 when it first sees $\Phi_e(A, D, x) \downarrow$. However, enumeration by σ into D can upset a bound appointed by τ, by destroying the computation $\Phi_e(A, D, x)$.
Suppose that we first see $\Phi_e(A, D, x) \downarrow$ at stage s, and $o^i_s(p) \downarrow$. Then we have a bound on the amount of times that σ will act for p, a bound for the amount

of times that $\Phi_e(A, D, x)$ can be destroyed by p. We can then account for this exact ordinal when appointing a bound to x.
If instead $o^i_s(p) \uparrow$, we cannot give an exact bound at this stage. However, we do know that if we eventually see a bound it must be less than ω. We can then use ω as an interim bound, until we eventually see $o^i(p) \downarrow$ and can update to the 'true' bound. It is this initial use of ω as a bound on action that requires an ω^3 bound on $\Phi_e(A, D)$, as opposed to ω^2.

Enumeration into A can also destroy a computation $\Phi_e(A, D, x)$; but A is provided by the opponent, and therefore beyond our control. To counter this, we use the fact that $\boldsymbol{a}$ is totally ω-c.a.. We build a shadow functional $\hat{\Phi}_e(A)$ to monitor change in A, ensuring that $\hat{\Phi}_e(A)$ is total if $\Phi_e(A, D)$ is total. We define, for each $x < \omega$, a tracker $c < \omega$ for which $\hat{\Phi}_e(A, c)$ has use $\varphi_e(x)$. Suppose that $\Phi_e(A, D, x)$ is destroyed by change in D. If A was correct up to the use of x, then c is now useless as a tracker. We immediately cancel c, and replace it with a new tracker when $\Phi_e(A, D)$ is next expansionary.
Since $\boldsymbol{a}$ is totally ω-c.a., if $\hat{\Phi}_e(A)$ is total, there must be some j such that $\hat{\Phi}_e(A) = f^j$ (the j^{th} member of the enumeration containing all ω-c.a. functions). As the strategy progresses, we (correctly) guess this j; then $o^j(c)$ bounds destruction caused by A-change against the computation $\Phi_e(A, D, x)$ while c remains the tracker of x.

Strategy Tree Construction:

Suppose τ works for requirement Q_e; then τ is responsible for building the shadow functional $\hat{\Phi}_\tau(A)$. The node τ has two outcomes, ∞ and f, ordered $\infty <$ f. This first tier below τ measures whether or not $\Phi_e(A, D)$ is expansionary at the current stage.
Then $\tau\hat{\ }\infty$ has outcomes ∞_i, f_i for all $i < \omega$, ordered $\infty_0 < \mathrm{f}_0 < \infty_1 < \cdots$, which guess whether $\hat{\Phi}_\tau(A)$ is total. To make this guess, we use the same trick employed in Theorem 2.1.3: we obtain a computable list $l_s(\tau, n)$ of se-

quences, non-decreasing in s, such that $\hat{\Phi}_\tau(A)$ is total if, and only if, there is an n such that the sequence $\langle l_s(\tau, n)\rangle$ is unbounded.
The node $\tau\hat{\ }\infty\hat{\ }\infty_n$, for any n, has outcomes $j < \omega$, ordered with the usual natural number ordering. This tier guesses for which j we have $\hat{\Phi}_\tau(A) = f^j$. The nodes $\tau\mathrm{f}$, $\tau\hat{\ }\infty\mathrm{f}_i$ for any i, and $\tau\hat{\ }\infty\hat{\ }\infty_n i$ for any n, i, all work for the next priority requirement.

A node σ working for requirement P^i has a unique outcome, which works for the next priority requirement. Suppose a follower p is appointed to σ at stage s. Unlike many of our other constructions, we have no mechanism by which to permit p; as such, we need to protect all computations already in place at stage s from action for p. We define $m^\tau(p)$, for all τ such that $\tau\hat{\ }\infty\hat{\ }\infty_n \preceq \sigma$ for some n, to indicate these computations.
Suppose τ works for requirement Q_e; we let $\tau \in \mathrm{prec}_\infty(\sigma)$ if $\tau\hat{\ }\infty\hat{\ }\infty_n \prec \sigma$ for some n, and let $\tau \in \mathrm{prec}_\mathrm{f}(\sigma)$ if $\tau\hat{\ }\infty\hat{\ }\mathrm{f}_y \preceq \sigma$ for some y. For $x < \omega$, *p may not injure* $\Phi_e(A, D, x)$ if τ works for Q_e, and either:

(i) $\tau \in \mathrm{prec}_\infty(\sigma)$, and $x < m^\tau(p)$; or,

(ii) $\tau \in \mathrm{prec}_\mathrm{f}(\sigma)$, and $x < y$.

Let the requirements be ordered in order type ω, and let s be a stage. Let γ_s denote the set of nodes accessible at stage s.

Let τ working for Q_e be accessible at stage s. Let $t < s$ be the last stage before s at which $\tau\hat{\ }\infty$ was accessible, or let $t = 0$ if there is no such stage. If $\mathrm{dom}\,\Phi_e(A, D)[s] \leqslant t$, let $\tau\hat{\ }\mathrm{f} \in \gamma_s$; otherwise, let $\tau\hat{\ }\infty \in \gamma_s$.

Suppose that we have $\tau\hat{\ }\infty \in \gamma_s$. Let y be least such that either $\Phi_e(A, D, y)\uparrow [t]$, or the computation $\Phi_e(A, D, y)[t]$ has been destroyed since stage t. For each $n < s$, let $t_n < s$ be the last stage before s at which $\tau\hat{\ }\infty\hat{\ }\infty_n$ was accessible, or $t_n = 0$ if there is no such stage. If there is an $n \leqslant y$ such that

$l_s(\tau, n) \geqslant t_n$, let $\tau\hat{\ }\infty\hat{\ }\infty_n \in \gamma_s$. Otherwise, let $\tau\hat{\ }\infty\hat{\ }\mathrm{f}_y \in \gamma_s$.

While $\tau\hat{\ }\infty$ is accessible, we maintain the shadow functional $\hat{\Phi}_\tau(A)$. Let $x < \omega$ be such that x has a defined tracker $c = \mathrm{tr}_s(\tau, x)$. If $\hat{\Phi}_\tau(A, c) \uparrow [s]$, and either

- $\tau\hat{\ }\infty\hat{\ }\mathrm{f}_y$ is next accessible, and $x < y$; or,
- $\tau\hat{\ }\infty\hat{\ }\infty_n$ is next accessible, and $x < t$

holds, we define $\hat{\Phi}_{\tau,s+1}(A_s, c) = s$ with use $\varphi_{e,s}(x)$. For any $c < s$, if c is not a tracker for any x at τ and $\hat{\Phi}_\tau(A, c) \uparrow [s]$, we define $\hat{\Phi}_{\tau,s+1}(A_s, c) = 0$ with use 0. Finally, for all $x < s$ with no current tracker, we define a new, large tracker $\mathrm{tr}_{s+1}(\tau, x)$.

Suppose $\tau\hat{\ }\infty\hat{\ }\infty_n \in \gamma_s$. For each $i < s$, let $r_i < s$ be the last stage prior to s at which $\tau\hat{\ }\infty\hat{\ }\infty_n i$ was accessible, or $r_i = 0$ if there is no such stage. We let $r_s = 0$. Let $\tau\hat{\ }\infty\hat{\ }\infty_n i \in \gamma_s$ for the least $i \leqslant s$ such that for all $x < r_i$, $c = \mathrm{tr}_s(\tau, x)$ is defined, $o^i_s(c) \downarrow$, and $\hat{\Phi}_\tau(A, c)[s] = f^i_s(c)$.

Let σ working for P^i be accessible at s, and let π be the unique child node of σ. One of several cases will apply to σ.

(i) σ has no follower. We appoint a new, large follower $p = p(\sigma, s+1)$. We then define $\Lambda(A, D, p)[s+1] = s+1$ with use $\lambda_{s+1}(p) = p$, and define, for all $\tau \in \mathrm{prec}_\infty(\sigma)$, the value $m^\tau(p) = \mathrm{dom}\Phi_d(A, D)[s]$ where τ works for Q_d. Finally, let $\pi \in \gamma_s$.

(ii) σ has a follower $p = p(\sigma, s)$, but $o^i_s(p) \uparrow$. We immediately let $\pi \in \gamma_s$.

(iii) σ has a follower $p = p(\sigma, s)$ and $o^i_s(p) \downarrow$. If $\Lambda(A, D, p)[s] \neq f^i_s(p)$, immediately let $\pi \in \gamma_s$. Otherwise $\Lambda(A, D, p)[s] = f^i_s(p)$, so we enumerate $\lambda_s(p)$ into D_{s+1} and redefine $\Lambda(A, D, p)[s+1] = s+1$ with large use. We then cancel trackers for $x \geqslant m^\tau(p)$ for all $\tau \in \mathrm{prec}_\infty(\sigma)$, initialise all weaker nodes, and end the stage.

At the end of stage s, we maintain our functional Λ. For any $p \leqslant s$ for which $\lambda(p)$ was not already redefined at s, if $p \notin \mathrm{dom}\Lambda_s(A_{s+1}, D_{s+1})$ we let $\Lambda(A, D, p)[s+1] = s+1$. If p is a follower of a node σ working for requirement P^i, let $\lambda_{s+1}(p) = \lambda_s(p)$; otherwise, let $\lambda_{s+1}(p) = 0$. For any follower $p = p(\sigma, s)$ not cancelled at stage s (by initialisation), let $p(\sigma, s+1) = p$.

Verification:

First, we ensure that any follower $p(\sigma, s)$ is prevented from injuring any of its protected computations.

LEMMA 2.2.2: *Let σ be a node working for requirement P^i, and let $p = p(\sigma, s)$ be a follower appointed at stage $r \leqslant s$ and not cancelled since. For every computation $\Phi_e(A, D, x)$ that p may not injure,*

$$\Phi_e(A, D, x)[r] \downarrow = \Phi_e(A, D, x)[s].$$

Proof. Certainly $\Phi_e(A, D, x) \downarrow$ at r, the stage of p's appointment, as this is the defining factor of a computation that p may not injure.
Suppose then that $\Phi_e(A, D, x)[r] \neq \Phi_e(A, D, x)[s]$. At some stage $t \in [r, s)$, some node σ' working for $P^{i'}$ acted for its follower $p' = p(\sigma', t)$, with use $\lambda_t(p) < \varphi_{e,t}(x) = \varphi_{e,r}(x)$. If σ' is stronger than or to the left of σ, then p would be cancelled at stage t. Hence, σ' is weaker than σ, and p' must have been appointed at a stage $r' \in [r, t)$.
If τ working for Q_e is in $\mathrm{prec}_\infty(\sigma)$, then since r, r' are both $\tau\hat{}\infty$-stages, $m^\tau(p') =$ dom $\Phi_e(A, D)[r'] \geqslant$ dom $\Phi_e(A, D)[r] = m^\tau(p)$. Then $\lambda_{r'}(p')$ is defined to be large; certainly $\lambda_{r'}(p') > \varphi_{e,t}(x)$.
If τ is instead in $\mathrm{prec}_{\mathrm{f}}(\sigma)$, there is a $y > x$ which is least such that $\Phi_e(A, D, y)$ was destroyed at or since the previous $\tau\hat{}\infty$-stage before r'. Hence, we have $\Phi_e(A, D, x) \downarrow [r']$, and we define $\lambda_{r'}(p') > \varphi_{e,t}(x)$.

□

Hence, if a protected computation $\Phi_e(A, D, x)$ (w.r.t. p) is destroyed, then p is cancelled. Then p is prevented from injuring the computation when it

later reconverges with a larger use.
We note that the construction is identical to that of Theorem 2.1.3 with regards to Q requirements and definition of Λ, and very similar with regards to P requirements. As such, we may borrow Corollary 2.1.5 and Lemmata 2.1.6, 2.1.7, 2.1.8, and 2.1.9 from the verification of Theorem 2.1.3 with at most superficial modification to their proofs. Thus, Λ is total, the true path is infinite, and the λ use of any follower p is always larger than the φ use of any computation it may not injure. We also have the following lemmata:

LEMMA 2.2.3: *Let τ be a node working for requirement Q_e. Let s be a stage; let $x < \omega$ be such that $c = tr_s(\tau, x)$ is defined. Suppose that $\hat{\Phi}_\tau(A, c) \downarrow [s]$, and let $u = \hat{\varphi}_{\tau,s}(c)$. Then:*

(i) $\Phi_e(A, D, x) \downarrow [s]$ *and* $u = \varphi_{e,s}(x)$*; and,*

(ii) If $D_s \upharpoonright u \neq D_{s+1} \upharpoonright u$*, then the tracker c is cancelled at s.*

LEMMA 2.2.4: *Let τ be the node on the true path working for Q_e. Then $\Phi_e(A, D)$ is total if and only if for some $n < \omega$, $\tau\hat{}\infty\hat{}\infty_n$ is on the true path, if and only if $\hat{\Phi}_\tau(A)$ is total.*

With these facts established, we now proceed to prove that the P and Q requirements are met.

LEMMA 2.2.5: *For every $i < \omega$, the requirement P^i is met.*

Proof. Fix $i < \omega$, and let σ be the node on the true path working for requirement P^i. Let s^* be the final stage at which σ is initialised, and $s > s^*$ be the next σ-stage. If there is no stage $t > s$ at which we see $o^i_t(p) \downarrow$ for $p = p(\sigma, s + 1)$, then P^i is met by false hypothesis. Suppose instead that there is such a stage. The node σ can act only finitely often for p, so let $t > s$ be the least stage such that σ does not act for p after stage t.
Suppose that $f^i(p) = \Lambda(A, D, p)[t + 1]$. At stage t we defined $\Lambda(A, D, p)[t + 1] = t+1 \neq f^i_t(p)$. Therefore $f^i(p) \neq f^i_t(p)$, so there is a stage $t' > t$ at which

we see $f^i_{t'}(p) = f^i(p)$. Since σ is on the true path, it must become accessible again at some stage $r \geqslant t'$; at this stage, we redefine $\Lambda(A, D, p)[r] \neq f^i(p)$, so P^i is met.

□

Lemma 2.2.6: *For every $e < \omega$, the requirement Q_e is met.*

Proof. Fix $e < \omega$, and let τ be the node on the true path working for Q_e; if $\Phi_e(A, D)$ is total, by Lemma 2.2.4 we have $\rho = \tau\hat{\ }\infty\hat{\ }\infty_n i$ on the true path for some n, i. Let s^* be the last stage at which ρ is initialised, and let $s_0 < s_1 < \cdots$ be the ρ-stages following s^*. We proceed to build functions g, m such that $\langle g_s, m_s\rangle_{s<\omega}$ is an ω^3-computable approximation for $\Phi_e(A, D)$.

Fix $x < \omega$. Let $i(x)$ be the least j such that $x <$ dom $\Phi_e(A, D)[s_{j-1}]$. Let $a(x)$ be the set of all pairs $\langle\sigma, p\rangle$ such that $\tau \in \mathrm{prec}_\infty(\sigma)$ or $\tau\hat{\ }\infty\hat{\ }\mathrm{f}_y \preceq \sigma$ for $y > x$, and p is a follower for σ that was appointed before, but not cancelled by, stage $s_{i(x)}$ for which we have $o^i_{s_{i(x)}}(p)\downarrow$. For each $j \geqslant i(x)$, we refine the set $a(x)$ to the set $a_j(x)$: let $a_j(x)$ be the set of all $\langle\sigma, p\rangle \in a(x)$ such that p has not been cancelled by stage s_j.
Similarly, let $b(x)$ be the set of all pairs $\langle\sigma, p\rangle$ such that $\tau \in \mathrm{prec}_\infty(\sigma)$ or $\tau\hat{\ }\infty\hat{\ }\mathrm{f}_y \preceq \sigma$ for $y > x$, and p is a follower for σ that was appointed before, but not cancelled by, stage $s_{i(x)}$ but for which $o^i_{s_{i(x)}}(p)\uparrow$ if σ works for P^i. For all $j \geqslant i(x)$, we also produce the set $b_j(x)$: let $b_j(x)$ be the set of all $\langle\sigma, p\rangle \in b(x)$ such that we still see $o^i_{s_j}(p)\uparrow$ if σ works for P^i and p has not been cancelled by stage s_j.
We see that the sets $a(x)$ and $b(x)$ together comprise all followers that are capable of injuring the computation $\Phi_e(AD, x)$. We note that if $\langle\sigma, p\rangle \in b_j(x)$ for some $j \geqslant i(x)$, and $o^i_{s_{j+1}}(p)\downarrow$ but p is not cancelled on the interval $[s_j, s_{j+1})$, then $\langle\sigma, p\rangle \in a_{j+1}(x)$.

Let $j \geqslant i(x)$, and let $\langle\sigma_0, p_0\rangle, \langle\sigma_1, p_1\rangle, \ldots, \langle\sigma_n, p_n\rangle$ be the members of $a_j(x)$. Suppose that for all $0 \leqslant k \leqslant n$, the node σ_k works for the requirement P^{i_k}. For each k, let $t_{k,j}$ be the last stage prior to s_j at which σ_k acted for p_k, or if there is no such stage, let $t_{k,j}$ be the stage at which p_k was appointed. Given that $\hat{\Phi}_\tau(A) = f^i$, we note that for any $j \geqslant i(x)$, we have $\Phi_e(A, D, x) \downarrow [s_j]$, $c_j = c_j(x) = \mathrm{tr}_{s_j}(\tau, x)$ is defined, $o^i_{s_j}(c_j) \downarrow$, and $\hat{\Phi}_\tau(A, c_j) \downarrow [s_j] = f^i_{s_j}(c_j)$.

We then define $g_j(x) = \Phi_e(A, D, x)[s_j]$, and letting $\eta_j(x) = o^{i_0}_{t_{0,j}}(p_0) + \ldots + o^{i_n}_{t_{n,j}}(p_n)$, we also define

$$m_j(x) = \omega^2 \cdot |b_j(x)| + \omega \cdot \eta_j(x) + o^i_{s_j}(c_j).$$

It is clear that $\lim_s g_s(x) = \Phi_e(A, D, x)$. For any j, x, the values $|b_j(x)|$, $o^{i_k}_{t_{k,j}}(p_k)$ for any $0 \leqslant k \leqslant n$, and $o^i_{s_j}(c_j)$ are all bounded by ω, and hence the sum $m_j(x)$ is bounded by ω^3.

Suppose that $g_j(x) \neq g_{j+1}(x)$. If this is caused by change in A alone, then $c_j = c_{j+1}$ and at $r \in [s_j, s_{j+1})$ there is some enumeration into A below $\varphi_{e,r}(x)$. By Lemma 2.2.3, this implies that $\hat{\Phi}_\tau(A, c_j) \uparrow [r + 1]$, but we know $\hat{\Phi}_\tau(A, c_j) \downarrow [s_j]$ and $\hat{\Phi}_\tau(A, c_j) \downarrow [s_{j+1}]$. Then $\hat{\Phi}_\tau(A, c_j)[s_j] \neq \hat{\Phi}_\tau(A, c_j)[s_{j+1}]$, so $f^i_{s_j}(c_j) \neq f^i_{s_{j+1}}(c_j)$. Therefore, $o^i_{s_j}(c_j) > o^i_{s_{j+1}}(c_j)$, and consequently $m_j(x) > m_{j+1}(x)$.

If the inequality is instead caused by change in D, then there is an enumeration into D below $\varphi_{e,s_j}(x)$ at stage s_j by some $(\sigma, p) \in a_j(x)$. Hence we must have $\Lambda(A, D, p)[s_j] = f^i_{s_j}(p)$. Let k be such that $\sigma = \sigma_k, p = p_k$. At stage $t_{k,j}$, we defined $\Lambda(A, D, p)[t_{k,j} + 1] = t_{k,j} + 1 > f^{i_k}_{t_{k,j}}(p)$. Hence from $f^{i_k}_{s_j}(p) = \Lambda(A, D, p)[s_j]$ we may conclude that $f^{i_k}_{t_{k,j}}(p) \neq f^{i_k}_{s_j}(p)$, and hence $o^{i_k}_{t_{k,j}}(p) > o^{i_k}_{s_j}(p)$. Since $t_{k,j+1} = s_j$, we then have $o^{i_k}_{t_{k,j}}(p) > o^{i_k}_{t_{k,j+1}}(p)$.
Since all summands of η are natural numbers, it then follows from $o^{i_k}_{t_{k,j}}(p) > o^{i_k}_{t_{k,j+1}}(p)$ that $\eta_j(x) > \eta_{j+1}(x)$, and further that $m_j(x) > m_{j+1}(x)$, assuming that there is no σ' stronger than σ such that $\langle \sigma', p' \rangle \in b_j(x)$ and $\langle \sigma', p' \rangle \in a_{j+1}(x)$.
Suppose instead that there is such a pair $\langle \sigma', p' \rangle$. Then $|b_{j+1}(x)| < |b_j(x)|$. As $|b_{j+1}(x)|$ is the factor on the ω^2 term of $m_{j+1}(x)$, this decrease 'outweighs'

any possible increase in $\eta_{j+1}(x)$, as $\omega \cdot \eta_{j+1}(x)$ is bounded by ω^2. Hence, $m_j(x) > m_{j+1}(x)$.

Therefore, $g_j(x) \neq g_{j+1}(x)$ implies that $m_j(x) > m_{j+1}(x)$. Finally, we require that m is non-increasing. Since o^i and o^{i_k} for all k are non-increasing, as is $|b_j(x)|$ w.r.t. j, we would only see $m_j(x) < m_{j+1}(x)$ if $c_j \neq c_{j+1}$. However, this implies that $D_{s_j} \upharpoonright u \neq D_{s_{j+1}} \upharpoonright u$ for $u = \varphi_{e,s_j}(x)$. As shown above, this results in either $|b_j(x)| > |b_{j+1}(x)|$ or $|b_j(x)| = |b_{j+1}(x)|$ and $\eta_j(x) > \eta_{j+1}(x)$, and hence $m_j(x) > m_{j+1}(x)$.

Therefore, $\langle g_s, m_s \rangle_{s<\omega}$ is an ω^3-computable approximation of $\Phi_e(A, D)$.

□

This concludes the proof of Theorem 2.2.1.

2.3. Above a Superlow c.e. Degree

Let A be a set; then A is *superlow* if $A' \equiv_{tt} \emptyset'$. A Turing degree $\boldsymbol{a}$ is superlow if some $A \in \boldsymbol{a}$ is superlow. It is a result of Schaeffer [10] that every superlow c.e. degree is array computable; thus, every superlow c.e. degree is totally ω-c.a..

Let $\langle \Phi_e \rangle$ be an enumeration of all consistent functionals. The *jump function* of a set A is a binary function $J^A(-,-)$ such that, for all e and n, $J^A(e,n)$ gives whether or not $\Phi_e(A,n)$ halts, the stage at which it halts, and the output. If A is superlow, then the jump function of A is ω-c.a..

Theorem 2.3.1 proves that above every superlow c.e. degree $\boldsymbol{a}$, there is a degree which is ω^2-c.a., but not ω-c.a.. This proof could be easily modified to produce a degree bounding $\boldsymbol{a}$ which is maximal α-c.a. for any $\alpha \geqslant \omega^2$; thus, for every $\omega^2 \leqslant \alpha \leqslant \varepsilon_0$, there is a degree bounding $\boldsymbol{a}$ which is totally α-c.a., and not totally γ-c.a. for any $\gamma < \alpha$. Hence, there is no collapse in any level of the hierarchy of totally α-c.a. degrees above any superlow c.e. degree.

THEOREM 2.3.1: *Let c.e. A be superlow, and let $\boldsymbol{a} = \deg_T(A)$. Then there is a c.e. degree $\boldsymbol{d} > \boldsymbol{a}$ which is maximal totally ω^2-c.a..*

Let $\langle A_s \rangle$ be a given computable enumeration of A. We proceed to build a c.e. set D such that $\boldsymbol{d} = \deg_{\mathrm{T}}(A \oplus D)$ is maximal totally ω^2-c.a..

Requirements:

Let $\langle \Phi_e \rangle_{e<\omega}$ be an enumeration of all consistent functionals. First, we must ensure that $\boldsymbol{d}$ is totally ω^2-c.a. with the requirements

$$\text{for all } e < \omega,\ Q_e : \text{If } \Phi_e(A, D) \text{ is total, then it is } \omega^2\text{-c.a..}$$

Further, we must ensure its maximality with the requirements

$$\text{for all } e, i < \omega,\ P_e^i : \text{Either } \Lambda_e(A, D, W_e) \neq f^{\omega^2,i}, \text{ or } W_e = \Delta_e^i(A, D),$$

where Λ_e and Δ_e^i are functionals enumerated by us, and $f^{\omega^2,i} = f^i = \lim_s f_s^i$ where $\langle \langle f_s^i, o_s^i \rangle_{s<\omega} \rangle_{i<\omega}$ is an effective list of partial ω^2-computable approximations.

Discussion:

To meet a requirement P_e^i in isolation, a node σ working for P_e^i must appoint a follower $p = p(\sigma, j)$. Thereafter, at any stage for which $\Lambda_e(A, D, W_e, p)[s] = f_s^i(p)$, σ enumerates $\lambda_{e,s}(p)$ into D_{s+1}. However, this enumeration could destroy a computation $\Phi_d(A, D, x)[s]$; any node working for Q_d that could be affected by enumeration at σ needs to know a bound on this action to meet its requirement. The strategy we employ to ensure this is the same as that which we use repeatedly in this text: we only allow p to injure $\Phi_d(A, D, x)$ if $o^i(p) \downarrow$ by the stage at which we first see $\Phi_d(A, D, x) \downarrow$. We then use $j \downarrow W_e$ to permit p, regardless of whether σ is accessible at that stage, allowing us to redefine $\lambda_e(p)$ to be large and commence an attack with p. We cannot guarantee that any one follower be permitted, so we continue to appoint followers if no current follower has been permitted.

Where we differ from Theorem 2.1.3 is in our handling of injury to $\Phi_d(A, D)$ computations caused by enumeration into A. The enumeration $\langle A_s \rangle$ of A is again beyond our control, necessitating the building of a shadow functional $\hat{\Phi}_\tau(A)$ at each node τ working for Q_d. In Theorem 2.1.3, we are allowed only one tracker for $x < \omega$ to be in place at any time. Otherwise, we run the risk that $\hat{\Phi}_\tau(A)$ ends up partial, which could prevent us from meeting Q requirements. However, here we have the advantage that A is superlow, and its jump function J^A is therefore ω-c.a.. Let $\langle f_s, o_s \rangle_{s<\omega}$ be an ω-computable approximation of J^A; there is then a d' known to us such that $\langle f_s(d', -), o_s(d', -) \rangle$ is an ω-computable approximation to $\hat{\Phi}_\tau(A)$. For simplicity we will drop reference to d', and refer to the approximation only as $\langle f_s, o_s \rangle$ for every τ, though remembering that d' will actually vary with τ.
For each $x < \omega$ we assign, using Cantor's pairing function, an infinite set $\{\pi(a, n) \mid n < \omega\}$ (also referred to as the column $\pi(a, -)$) of natural numbers from which we select trackers for x. From this column, at stage s we *reserve* as many trackers as there are foreseeable injuries caused by D-change. At stages progress, we may change the number of reserved trackers as we see updated predictions of injuries. Each enumeration into D by a follower p of a node σ working for a P-requirement cancels the current tracker of x.
For the current tracker c of x, we define $\hat{\Phi}_\tau(A, c)[s + 1] = s + 1$ so as to intentionally differ from $f_s(c)$. The number of times that $f(c)$ will change value is bounded by $o(c)$, and $\lim_s f_s(c) = \hat{\Phi}_\tau(A, c)$; since we define $\hat{\Phi}_\tau(A, c)$ to have the same use as $\Phi_d(A, D, x)$, the value $o(c)$ then also bounds the number of times $\Phi_d(A, D, x)$ will be destroyed by A-change while c remains a tracker of x, as long as we only believe the computation $\Phi_d(A, D, x)$ (i.e. define the ω^2-computable approximation to $\Phi_d(A, D)$ on x) at stages where $\hat{\Phi}_\tau(A, c) = f(c)$.
Suppose however that c is the tracker of x at stage s, and c' the tracker of

x at stage $t > s$; it is entirely possible that $o_s(c) < o_t(c')$. If $\langle \hat{f}_s, \hat{o}_s \rangle_{s<\omega}$ is the approximation to $\Phi_d(A, D)$ we are building, then including $o_s(c)$ as a summand of $\hat{o}_s(x)$ for only the current tracker is not conducive to fulfilling the non-increasing requirement imposed on $\hat{o}$. Hence, at stage s, we use the sum of $o_s(c)$ for *all* reserved trackers from column $\pi(a, -)$.

Strategy Tree Construction:

Every node on the tree, whichever requirement it works for, has two outcomes $\infty <$ f. For a node τ working for a requirement Q_e, these outcomes respectively measure whether or not $\Phi_e(A, D)$ is expansionary when τ is visited. The node $\tau\hat{}\infty$ has outcomes $\imath < \emptyset$, which measure whether or not $\hat{\Phi}_\tau(A)$ appears to agree with f at that stage. If $\Phi_e(A, D)$ is total, and τ is on the true path, then $\tau\hat{}\infty\hat{}\imath$ will clearly be on the true path.
For a node σ working for a requirement P^i_e, the outcomes ∞, f measure whether we believe σ will not, or will, succeed in completing an attack with a follower p, and force $\Lambda_e(A, D, W_e, p) \neq f^i(p)$. Both outcomes of σ, both outcomes of $\tau\hat{}\infty$, and $\tau\hat{}$f all work for the next (lower) priority requirement.

Order the requirements in order type ω, and let s be a stage. Let γ_s denote the set of accessible nodes at stage s.

First at s: is there a node σ working for a requirement P^i_e with a realised follower $p = p(\sigma, j)$, such that $j \in W_{e,s} \setminus W_{e,s-1}$? If there is, take the strongest such σ, cancel all followers p' of σ such that $p' > p$, and redefine $\Lambda_e(A, D, W_e, p)[s+1] = s+1$ with large use $\lambda_{e,s+1}(p)$. We initialise all nodes weaker than $\sigma\hat{}$f, inclusive, and repeat this process until no further followers may be permitted.

If $\tau \in \gamma_s$ works for Q_e, let $t < s$ be the last stage at which $\tau\hat{}\infty$ was accessible, or $t = 0$ if there was no such stage. If dom $\Phi_e(A, D)[s] < t$, let $\tau\hat{}$f $\in \gamma_s$.

Otherwise, let $\tau\hat{}\infty \in \gamma_s$.
While $\tau\hat{}\infty$ is accessible, we assign new trackers and maintain $\hat{\Phi}_\tau(A)$. For all $x < s$, if x is not assigned a column $\pi(a, -)$ at τ, then assign to x the least (w.r.t. a) unassigned column. If x has no current tracker, and $\pi(a, -)$ is its column, let b be least such that $\pi(a, b)$ has not previously been a tracker for x; we appoint $\pi(a, b)$ to be the new tracker of x at τ by setting $\mathrm{tr}_{s+1}(\tau, x) = \pi(a, b)$.
Let $c = \mathrm{tr}_s(\tau, x)$ be the current tracker for x at τ at stage s. If $\hat{\Phi}_\tau(A, c) \uparrow [s]$, we redefine $\hat{\Phi}_\tau(A_s, c)[s + 1] = s + 1$ with use $\varphi_{e,s}(x)$.

If $\tau\hat{}\infty \in \gamma_s$, let $r < s$ be the last stage at which $\tau\hat{}\infty\hat{}1$ was accessible. We let $\tau\hat{}\infty\hat{}1 \in \gamma_s$ if, for all $x < r$, $c = \mathrm{tr}_s(\tau, x)$ is defined, $o_s(c) \downarrow$, and $\hat{\Phi}_\tau(A, c)[s] = f_s(c)$. Otherwise, let $\tau\hat{}\infty\hat{}\emptyset \in \gamma_s$.

If $\sigma \in \gamma_s$ works for P^i_e, and:

(i) σ has permitted follower $p = p(\sigma, j)$.

   (a) If $\Delta_\sigma(A, D, j) \uparrow [s]$, cancel the follower p, and set $\Delta_\sigma(A, D, j) = 1$ with use 0. We then let $\sigma\hat{}\infty$ be next accessible.

   (b) If $\Delta_\sigma(A, D, j) \downarrow [s]$ and $\Lambda_e(D, W_e, p)[s] \neq f^i_s(p)$, we let $\sigma\hat{}\mathrm{f}$ be next accessible.

   (c) If $\Delta_\sigma(A, D, j) \downarrow [s]$ and $\Lambda_e(D, W_e, p)[s] = f^i_s(p)$, we enumerate $\lambda_{e,s}(p)$ into D_{s+1}, and redefine $\Lambda_e(D, W_e, p)[s + 1] = s + 1$ with large use. We then initialise all nodes weaker than $\sigma\hat{}\mathrm{f}$ (inclusive), and cancel the least tracker for each $x \geqslant m^\tau(p)$ for all $\tau \in \mathrm{prec}_\infty(\sigma)$. We then end the stage.

(ii) σ has a follower p such that $o^i_s(p) \uparrow$. We immediately let $\sigma\hat{}\mathrm{f}$ be next accessible.

(iii) For every follower p of σ, $o^i_s(p) \downarrow$ but p is not permitted. Let $t < s$ be the last stage at which σ was accessible, if it exists. If for a fol-

lower $p = p(\sigma, j)$ of σ we have either: (a) $o_t^i(p) \uparrow$ while $o_s^i(p) \downarrow$, or (b) $\Delta_\sigma(A, D, j) \uparrow [s]$, we proceed to define $\Delta_\sigma(A, D, j)[s+1] = W_{e,s+1}(j)$. If $W_{e,s}(j) = 1$, we set the use of this computation to be 0 and cancel p. If (a) holds and if $W_{e,s}(j) = 0$, then this is the stage at which p becomes realised. Then for each $\tau \in \text{prec}_\infty(\sigma)$, if τ works for Q_d we define $m^\tau(p) = \text{dom}\Phi_d(A, D)[s]$.
If $W_{e,s}(j) = 0$, then set the use of $\Delta_\sigma(A, D, j)[s+1]$ to be:
$\delta_{\sigma,s+1}(j) = \max\{\varphi_{d,s}(x) : p \text{ may not injure } \Phi_d(A, D, x)\} + 1$.

Let k be largest such that $p(\sigma, k)$ is a follower of σ. Let $m > k$ be least such that $m \notin W_{e,s}$. For any $k < n < m$, if $\Delta_\sigma(A, D, n) \uparrow [s]$, define $\Delta_\sigma(A, D, n)[s+1] = 1$ with use 0. We then appoint a new, large follower $p(\sigma, m)$, define $\Lambda_e(D, W_e, p)[s+1] = s+1$ with use $\lambda_{e,s+1}(p) = p$, and let $\sigma\hat{}\infty$ be next accessible.

At the conclusion of stage s, we maintain the functional Λ to ensure its totality. For any pair $(e, p) \leqslant s$ for which $\lambda_e(p)$ was not already redefined at s, if $p \notin \text{dom } \Lambda_{e,s}(D_{s+1}, W_{e,s+1})$ we let $\Lambda_e(D, W_e, p)[s+1] = s+1$. If p is a follower of a node σ working for requirement P_e^i, let $\lambda_{e,s+1}(p) = \lambda_{e,s}(p)$; otherwise, let $\lambda_{e,s+1}(p) = 0$.

Verification:

Due to the similarity of this construction to that of Theorem 2.1.3, we may borrow several lemmata from its verification with at most minor modification. Namely, Lemmata 2.1.4, 2.1.6, 2.1.8, 2.1.9, and 2.1.10 all hold. It only remains to be shown that our construction meets the Q requirements.

LEMMA 2.3.2: *For all $e < \omega$, the requirement Q_e is met.*

Proof. For fixed $e < \omega$, suppose that $\Phi_e(A, D)$ is total, and let τ be the node on the true path working for Q_e; then $\tau\hat{}\infty\hat{}1$ is also on the true path. Let s^* be the last stage at which τ is initialised, and let $s_0 < s_1 < \cdots$ be the stages

after s^* at which $\tau\hat{\ }\infty 1$ is accessible. We proceed to build an ω^2-computable approximation $\langle \hat{f}_s, \hat{o}_s\rangle_{s<\omega}$ for $\Phi_e(A, D)$.
Fix $x < \omega$. We define $i(x)$ to be the least j such that $x <$ dom $\Phi_e(A, D)[s_{j-1}]$, and define a set $a(x)$ such that $\langle \sigma, p\rangle \in a(x)$ if, and only if, σ works for some requirement P^i_d, and p is a follower of σ appointed and realised but not cancelled prior to stage $s_{i(x)}$. For all $j \geqslant i(x)$, we define $a_j(x)$ such that $\langle \sigma, p\rangle \in a_j(x)$ if, and only if, $(\sigma, p) \in a(x)$ and p is not cancelled prior to s_j.

We note that for every $j \geqslant i(x)$, $c = \mathrm{tr}_{s_j}(\tau, x)$ is defined, $o_{s_j}(c)\downarrow$, and $\hat{\Phi}_\tau(A, c)[s_j] = f_{s_j}(c)$.
For $j \geqslant i(x)$, we define $\hat{f}_j(x) = \Phi_e(A, D, x)[s_j]$.
Let $\langle \sigma_1, p_1\rangle, \ldots, \langle \sigma_n, p_n\rangle$ be the members of $a_j(x)$, with each σ_l working for requirement $P^{i_l}_{d_l}$. For each $1 \leqslant l \leqslant n$, let $t_{l,j}$ be the last stage before stage s_j at which σ_l acted for p_l, including granting permission; if there is no such stage, then let $t_{l,j}$ be the stage at which p_l was appointed. Let $m_{l,j}, k_{l,j}$ be the natural numbers such that $o^{i_l}_{t_{l,j}}(p_l) = \omega \cdot m_{l,j} + k_{l,j}$. Using these, we then define the sums $m_j = m_{1,j} + \cdots + m_{n,j}$ and $k_j = k_{1,j} + \cdots + k_{n,j}$.
Suppose that $c = \pi(a, b)$. If $j = i(x)$, or if $j > i(x)$ and $k_j > k_{j-1}$, we define the set $C_j(x) = \{\pi(a, b') \mid b \leqslant b' \leqslant b + k_j\}$. If instead $k_j \leqslant k_{j-1}$, we define $C_j(x) = C_{j-1}(x)$.
We then define $\hat{o}_j(x) = \omega \cdot m_j + \Sigma_{c' \in C_j(x)} o_{s_j}(c') + k_j$.

It is clear that $\lim_s \hat{f}_s(x) = \Phi_e(A, D, x)$, and as all m_j, k_j, and $o_{s_j}(c')$ are natural numbers, $\hat{o}_j(x)$ is bounded by ω^2.

To show that $\hat{o}_s(x)$ is non-increasing with respect to s, it suffices to show that if $\Sigma_{c' \in C_j(x)} o_{s_j}(c') < \Sigma_{c' \in C_{j+1}(x)} o_{s_{j+1}}(c')$, then $m_j > m_{j+1}$. Suppose the antecedent of this implication to be true for some $j \geqslant i(x)$; then $k_{j+1} > k_j$, otherwise $C_{j+1}(x) = C_j(x)$ and we would have a contradiction. We must then have $k_{l,j+1} > k_{l,j}$ for some l, but since o is non-increasing, this implies that $m_{l,j+1} < m_{l,j}$ which in turn implies $m_{j+1} < m_j$.

Finally, we must show that if $\hat{f}_j(x) \neq \hat{f}_{j+1}(x)$, then $\hat{o}_j(x) > \hat{o}_{j+1}(x)$. Suppose $\Phi_e(A, D, x)[s_j] \neq \Phi_e(A, D, x)[s_{j+1}]$, and let $u = \varphi_{e,s_j}(x)$.

Suppose that $D_{s_j} \upharpoonright u \neq D_{s_{j+1}} \upharpoonright u$. Then there is a pair $\langle \sigma, p\rangle = \langle \sigma_l, p_l\rangle \in a_j(x)$ (working for $P^i_d = P^{i_l}_{d_l}$) such that, at stage $s_j = t_{l,j+1}$, σ enumerates $\lambda_{d,s_j}(p) < u$ into D_{s_j+1} because $\Lambda(A, D, W_d, p)[s_j] = f^i_{s_j}(p)$.

At stage $t_{l,j}$, we previously acted for p, and redefined $\Lambda(A, D, W_d, p)[t_{l,j}+1] = t_{l,j} + 1 > f^i_{t_{l,j}}(p)$. Then $f^i_{t_{l,j}}(p) \neq f^i_{s_j}(p) = f^i_{t_{l,j+1}}(p)$, so $o^i_{t_{l,j}}(p) > o^i_{t_{l,j+1}}(p)$. Either $k_{l,j} > k_{l,j+1}$, or $k_{l,j} \leqslant k_{l,j+1}$ and $m_{l,j} > m_{l,j+1}$; in either case, $\hat{o}_j(x) > \hat{o}_{j+1}(x)$.

Suppose instead that $A_{s_j} \upharpoonright u \neq A_{s_{j+1}} \upharpoonright u$, but D does not change below u on this interval. Then at some stage $r \in [s_j, s_{j+1})$, a number below u enters A. Let c be the tracker of x at τ at stage s_j; then c persists until stage s_{j+1}. By Lemma 2.1.6, $u = \hat{\varphi}_{\tau,s_j}(c)$, so the computation $\hat{\Phi}_\tau(A, c)[s_j]$ is also destroyed at r. Hence, $\hat{\Phi}_\tau(A, c)[s_j] \neq \hat{\Phi}_\tau(A, c)[s_{j+1}]$, and therefore $f_{s_j}(c) \neq f_{s_{j+1}}(c)$. Then $o_{s_j}(c) > o_{s_{j+1}}(c)$ and certainly $c \in C_j(x)$ and $c \in C_{j+1}(x)$; consequently, $\hat{o}_j(x) > \hat{o}_{j+1}(x)$.

Hence, $\langle \hat{f}_s, \hat{o}_s\rangle_{s<\omega}$ is an ω^2-computable approximation for $\Phi_e(A, D)$, and Q_e is met.

$\square$

This concludes the proof of Theorem 2.3.1.

2.4. A Minimal Cover and a Cofinal Chain

We refer to the following theorem as producing a *minimal cover* in the hierarchy. What we mean is that we construct a pair $\boldsymbol{a}, \boldsymbol{d}$ of c.e. degrees such that every totally ω-c.a. c.e. degree above $\boldsymbol{a}$ is bounded by the totally ω-c.a. degree $\boldsymbol{d}$. This is not unique to ω; this proof can, with minor modification, produce such a pair for every $\alpha \leqslant \varepsilon_0$.

Theorem 2.4.1: *There are c.e. degrees $\boldsymbol{a} < \boldsymbol{d}$ such that $\boldsymbol{d}$ is totally ω-c.a. and for any degree $\hat{\boldsymbol{a}} > \boldsymbol{a}$, if $\hat{\boldsymbol{a}}$ is totally ω-c.a., then $\hat{\boldsymbol{a}} \leqslant \boldsymbol{d}$.*

We proceed to enumerate c.e. sets A and B such that $\deg_{\mathrm{T}}(A) = \boldsymbol{a}$ and $\deg_{\mathrm{T}}(A \oplus B) = \boldsymbol{d}$.

Requirements:

Let $\langle \Psi_e \rangle_{e<\omega}, \langle \Phi_e \rangle_{e<\omega}$ each be an enumeration of all consistent functionals. Firstly, we must ensure that $A \not\geqslant_{\mathrm{T}} B$, and thus $A <_{\mathrm{T}} A \oplus B$, by meeting the following requirements:

$$\text{for all } e < \omega,\ R_e : \Psi_e(A) \neq B.$$

We also require that $A \oplus B$ is totally ω-c.a., and achieve this with the set of requirements

$$\text{for all } e < \omega,\ Q_e : \text{If } \Phi_e(A, B) \text{ is total, then it is } \omega\text{-c.a.}$$

We must lastly ensure that $\deg_{\mathrm{T}}(A \oplus B)$ is indeed a minimal cover. To this end we require, for all $e < \omega$, that either $W_e \leqslant_{\mathrm{T}} A \oplus B$ or $\deg_{\mathrm{T}}(A \oplus W_e)$ is not totally ω-c.a..
Let $\langle \langle f^i_s, o^i_s \rangle_{s<\omega} \rangle_{i<\omega}$ be an effective list of partial ω-computable approximations such that letting $f^i = \lim_s f^i_s$, the sequence $\langle f^i \rangle_{i<\omega}$ contains every ω-c.a. function. As stated in the proof of Theorem 2.1.3, we assume that for any ω-c.a. function f, there is an i such that $f = f^i$, and for this i we have $\bigcup_s \operatorname{dom} o^i_s = \omega$. We then aim to meet the requirements

$$\text{for all } i, e < \omega,\ P^i_e : \text{If } \bigcup_s \operatorname{dom} o^i_s = \omega, \text{ either } \Lambda_e(A, W_e) \neq f^i \\ \text{or } W_e \leqslant_T A \oplus B.$$

Discussion:

To meet the P, Q, R requirements, we build a strategy tree.
In isolation, we meet an R_e requirement by choosing a follower x. We then wait until we see $\Psi_e(A, x)[s] = 0 = B_s(x)$. If this never happens, then $\Psi_e(A)$ and B disagree on x without our intervention; if it does occur, then we enumerate x into B_{s+1} and force a disagreement by freezing the computation

$\Phi_e(A, x)[s]$.

To meet a requirement Q_e when $\Phi_e(A, B)$ is total we must, for each $x < \omega$, associate x with an ordinal bound $n < \omega$ upon first seeing $\Phi_e(A, D, x) \downarrow$. Of course, this ordinal n must correctly bound the number of changes that the computation $\Phi_e(A, D, x)$ will undergo.

To meet requirement P_e^i in isolation, we select a follower p. At any stage where we see $\Lambda_e(A, W_e, p)[s] = f^i(p)[s]$, we enumerate $\lambda_{e,s}(p)$ into A_{s+1}.

However, in trying to meet these simultaneously, it is clear that enumeration into A or B for the sake of a P or R requirement may destroy a computation $\Phi_e(A, B, x)$; we then require that a node working for Q_e must be able to account for this action when it appoints an ordinal bound to $\Phi_e(A, B, x)$. Our strategy here is similar to that of Theorem 2.1.3, but simplified by the fact that any injury to $\Phi_e(A, B, x)$ must be caused by us, and not by an opponent-played set.

Suppose we first see $\Phi_e(A, B, x) \downarrow$ at stage s. If a node ρ working for R_d already has an appointed follower $y = y(\rho, s)$, we can allow ρ to act for y and injure $\Phi_e(A, B, x)$ as we know it will do so at most once. In a similar fashion, if a node σ working for P_d^i has a follower $p = p(\sigma, j)$ at s such that $o_s^i(p) \downarrow$, we can allow σ to injure $\Phi_e(A, B, x)$ when acting for p. This is because $o_s^i(p)$ is a bound on σ acting for p, and can be taken into account when appointing the bound to $\Phi_e(A, B, x)$ for Q_e. If $o_s^i(p) \uparrow$ instead, then we cannot immediately provide a bound for action by σ with respect to p and must prevent p from injuring the computation $\Phi_e(A, B, x)$.

For the sake of totality of Λ_d, we must define $\lambda_d(p)$ at the stage at which we appoint p. However, if τ working for Q_e certifies a computation $\Phi_e(A, B, x)$ after we appoint p but before $o_s^i(p) \downarrow$, then $\lambda_d(p)$ may be small enough to injure $\Phi_e(A, B, x)$. Since o^i may be partial, we cannot allow τ to wait for $o^i(p) \downarrow$ before appointing a bound to $\Phi_e(A, B, x)$. We can, however, prevent p from causing injury to $\Phi_e(A, B, x)$ if we are able to redefine the use $\lambda_{d,t}(p)$ to be large before σ ever acts for p, at a stage at which $\Phi_e(A, D, x) \downarrow [t]$.

Before we see $o^i(p) \downarrow$, σ does not need to act for p, so there is no conflict

in waiting until we see $o^i(p)\downarrow$ to redefine $\lambda_d(p)$. We then use $j\downarrow W_e$ to permit σ to increase the use $\lambda_d(p)$ to be large, allowing σ to attack with p without causing injury to $\Phi_e(A,B,x)$. Since we cannot guarantee that any one follower will be permitted, we appoint a sequence of followers and wait until one of them is permitted. We only allow a follower p to be permitted if $\Phi_e(A,B,x)\downarrow$ for all $\langle e,x\rangle$ such that p may not injure $\Phi_e(A,B,x)$. The new, lifted use $\lambda_d(p)$ is then too large to cause injury to any computation protected from p.
We note that the stage at which we see $j\downarrow W_e$ may not be a σ-stage, but we need to ensure that $\lambda_d(p)$ is redefined, and we cannot guarantee that σ will be accessible again. Hence, we carry out the permission whether or not it is a σ-stage.

Strategy Tree Construction:

Let τ be a node working for requirement Q_e. We give τ two child nodes, ∞ and f, with $\infty <$ f. Both child nodes work for the next lower priority requirement, with $\tau\hat{\ }\infty$ guessing that $\Phi_e(A,B)$ is total by being accessible at expansionary stages.

Let σ be a node working for P^i_e. Then σ has only one child node, working for the next requirement in the priority ordering. Suppose τ works for Q_d. We define $\tau\in\text{prec}(\sigma)$ if $\tau\hat{\ }\infty\preceq\sigma$.
Suppose a follower $p=p(\sigma,j)$ is appointed at some stage s; p is initially *unrealised*, but if t is the least stage after s at which we see $o^i_t(p)\downarrow$, then we refer to p as *realised* at and after the next σ-stage $t'>t$. At stage t', for every $\tau\in\text{prec}(\sigma)$ we define the value $m^\tau(p)$. The intent of $m^\tau(p)$, as in Theorem 2.1.3, is that for all $x<m^\tau(p)$, if τ works for Q_d then p is not allowed to injure the computation $\Phi_d(A,B,x)$. We may also refer to such a $\Phi_d(A,B,x)$ as a *protected computation* (with respect to p).
If p is a realised follower at stage t and we see $j\downarrow W_{e,t}$, if all protected com-

putations with respect to p halt at stage t, we may increase the use $\lambda_e(p)$ to be large. If this occurs, we refer to p as *permitted* from stage t.

Let ρ be a node working for R_e. Like σ above, ρ has a unique child node working for the next priority requirement, and if τ works for Q_d we define $\tau \in \text{prec}(\rho)$ if $\tau\hat{}\infty \preceq \rho$.
Suppose a follower $y = y(\rho, s)$ is appointed at stage s. At stage s, for every $\tau \in \text{prec}(\sigma)$ we define $m^\tau(y)$ to serve the same function as $m^\tau(p)$ does for a follower p of σ.

Let all P, Q, R requirements be ordered in order type ω, and let s be a stage. First at s, we check for any followers that may be permitted. We search for a node σ working for P_e^i such that:

(i) σ has a realised follower $p = p(\sigma, j)$;

(ii) $j \in W_{e,s} \setminus W_{e,s-1}$; and,

(iii) For all $\tau \in \text{prec}(\sigma)$, if τ works for Q_d, then $m^\tau(p) \leqslant \text{dom}\ \Phi_d(A, B)[s]$.

If there is such a node, we take the strongest such σ, cancel all followers of σ other than p, and redefine $\lambda_{e,s+1}(p)$ to be large. We then initialise all nodes weaker than σ.

Let γ_s be the set of nodes accessible at stage s. We define this set recursively as follows.

Let $\rho \in \gamma_s$ work for requirement R_e, and let π be the unique child node of ρ.

(i) ρ has no follower. We appoint a new, large follower $y = y(\rho, s+1)$ to ρ. For all $\tau \in \text{prec}(\rho)$, if τ works for Q_d, we define $m^\tau(y) = \text{dom}\Phi_d(A, B)[s]$. We then initialise all weaker nodes, and end the stage.

(ii) ρ has a follower $y = y(\rho, s)$, and $\Psi_e(A, y)[s] \neq B_s(y)$. We let $\pi \in \gamma_s$.

(iii) ρ has a follower y, and $\Psi_e(A, y)[s] = B_s(y)$. We enumerate y into B_{s+1}, initialise all weaker nodes, and end the stage.

Let $\tau \in \gamma_s$ work for Q_e. Let $t < s$ be the last stage before s at which $\tau\hat{}\infty \in \gamma_t$, or $t = 0$ if there is no such stage. If $\mathrm{dom}\Phi_e(A, B)[s] \leqslant t$, let $\tau\hat{}\mathrm{f} \in \gamma_s$; otherwise, let $\tau\hat{}\infty \in \gamma_s$.

Let $\sigma \in \gamma_s$ work for P^i_e, and let π be the child node of σ.

(i) σ has a permitted follower $p = p(\sigma, j)$. If $\Lambda_e(A, W_e, p)[s] \neq f^i_s(p)$, we let $\pi \in \gamma_s$.
Otherwise $\Lambda_e(A, W_e, p)[s] = f^i_s(p)$, so we enumerate $\lambda_{e,s}(p)$ into A_{s+1}. We then redefine $\Lambda_e(A, W_e, p)[s + 1] = s + 1$ with large use, initialise all weaker nodes and end the stage.

(ii) σ has a follower $p = p(\sigma, j)$ such that $o^i_s(p) \uparrow$. We immediately let $\pi \in \gamma_s$.

(iii) For every follower p of σ, we have $o^i_s(p) \downarrow$ but p is not permitted. This includes if σ has no followers, in which case we immediately appoint a new follower.
Let $t < s$ be the last stage such that $\sigma \in \gamma_t$. For the most recently appointed follower p, if $o^i_t(p) \uparrow$ then for each $\tau \in \mathrm{prec}(\sigma)$, if τ works for Q_d we define $m^\tau(p) = \mathrm{dom}\ \Phi_d(A, B)[s]$.
Let k be largest such that $p(\sigma, k)$ is a follower of σ. Let $m > k$ be least such that $m \notin W_{e,s}$. We then appoint a new, large follower $p(\sigma, m)$, define $\Lambda_e(A, W_e, p)[s+1] = s+1$ with use $\lambda_{e,s+1}(p) = p$, and let $\pi \in \gamma_s$.

When the stage is ended, we maintain the functional Λ to ensure that it is total. For any pair $(e, p) \leqslant s$ for which $\lambda_e(p)$ was not already redefined at s, if $p \notin \mathrm{dom}\ \Lambda_{e,s}(A_{s+1}, W_{e,s+1})$ we let $\Lambda_e(A, W_e, p)[s + 1] = s + 1$. If p is a follower of a node σ working for requirement P^i_e, let $\lambda_{e,s+1}(p) = \lambda_{e,s}(p)$; otherwise, let $\lambda_{e,s+1}(p) = 0$.

For every follower $y(\rho, s)$ not cancelled at stage s, let $y(\rho, s+1) = y(\rho, s)$.

Verification:

To begin, we show that the construction is fair to nodes on the true path, and that the true path γ_ω is therefore infinite.

LEMMA 2.4.2: *Let π be a node on the true path. Then π is initialised only finitely often.*

Proof. Suppose the lemma is true for all $\xi \prec \pi$, and suppose that ν is the parent node of π. Let s be the last stage at which ν is initialised. For action at ν after s to initialise π, ν must work for either a P or R requirement.
Suppose ν works for requirement R_e. Then ν will only initialise π when it appoints a follower $y = y(\nu, t)$ at the next ν-stage $t > s$, and if it later enumerates this follower into B.
Suppose instead that ν works for P^i_e. Then π will be initialised if ν permits a follower $p = p(\nu, j)$, and if ν subsequently enumerates $\lambda_{e,t}(p)$ into A_{t+1} for some $t > s$. After p is permitted, ν enumerates into A whenever we see $\Lambda_e(A, W_e, p)[t] = f^i_t(p)$, and we then redefine $\Lambda_e(A, W_e, p)[t+1] = t+1 \neq f^i_t(p)$. Since f^i is ω-c.a., the amount of times this can occur is bounded by some finite $n < \omega$.

□

Now that we know that the true path contains a node for every P, Q, and R requirement, we proceed to prove that each such node does indeed ensure the satisfaction of its associated requirement. We first handle the R requirements, which are the simplest.

LEMMA 2.4.3: *For all $e < \omega$, the requirement R_e is met.*

Proof. Fix e, and let ρ be the node on the true path working for R_e. By Lemma 2.4.2, let s^* be the last stage at which ρ is initialised. At the next ρ-stage $s > s^*$, ρ is appointed a follower $y = y(\rho, s+1)$ which is never cancelled.

If $\Psi_e(A, y) = B(y)$, then there is a stage t^* such that for all $t > t^*$, we have $\Psi_e(A, y)[t] = 0 = B_t(y)$. Since ρ is on the true path, there is a least ρ-stage $t > t^*$. At this stage, we enumerate y into B_{t+1}, so $B_{t+1}(y) = B_t(y) = 1$. We then initialise all weaker nodes, so any followers they appoint after stage t are too large to injure the computation $\Psi_e(A, y)[t]$. All nodes stronger than or to the left of ρ have finished acting by stage s; therefore $\Psi_e(A, y)[t] = \Psi_e(A, y) \neq B(y)$.

$\square$

For the lemmata that follow, we now define several helpful apparatus. Fix $x < \omega$, and suppose $\tau\hat{}\infty$ is on the true path, where τ works for Q_e. Let s^* be the last stage at which $\tau\hat{}\infty$ is initialised, and let $s_0 < s_1 < s_2 < \cdots$ be the stages after s^* at which $\tau\hat{}\infty$ is accessible.

Let $i(x)$ be the least i such that $x < \text{dom } \Phi_e(A, B)[s_i]$. Let $a(x)$ be the set of all pairs $\langle \sigma, p \rangle$ such that σ works for a P^i_d requirement, $\tau \in \text{prec}(\sigma)$, and p is a follower of σ realised prior to stage $s_{i(x)}$ but not cancelled by $s_{i(x)}$. For each $j \geqslant i(x)$, let $a_j(x)$ be the set of all pairs $\langle \sigma, p \rangle \in a(x)$ such that p is not cancelled by stage s_j.

Similarly, let $b(x)$ be the set of all pairs $\langle \rho, y \rangle$ such that ρ works for an R_d requirement, $\tau \in \text{prec}(\rho)$, and y is a follower of ρ appointed prior to $s_{i(x)}$ but not cancelled by stage $s_{i(x)}$. For each $j \geqslant i(x)$, let $b_j(x)$ be the set of all pairs $\langle \rho, y \rangle \in a(x)$ such that y is not cancelled by stage s_j, and $y \notin B_{s_j}$.

Lemmata 2.4.4 and 2.4.5 show that $a(x)$ and $b(x)$ together comprise all pairs of nodes and followers that are capable of causing injury to the computation $\Phi_e(A, B, x)$.

Lemma 2.4.4: *Let τ be a node working for Q_e. Let $\sigma \succeq \tau\hat{}\infty$ be a node working for P^i_d, and let p be a follower of σ such that p is permitted by stage $s \geqslant s_{i(x)}$. Suppose that $\langle \sigma, p \rangle \notin a(x)$. Then:*

(i) $m^\tau(p) > x$;

(ii) *Let t be the stage at which p became permitted. Then $x \in$ dom $\Phi_e(A, B)[t]$ and $A_t \oplus B_t \restriction \varphi_{e,t}(x) = A_s \oplus B_s \restriction \varphi_{e,t}(x)$;*

(iii) $\lambda_{d,s}(p) > \varphi_{e,s}(x)$.

Proof. (i) Let r^* be the stage at which p is realised. If $r^* < s_{i(x)}$, we would have $\langle \sigma, p \rangle \in a(x)$; hence $r^* \geqslant s_{i(x)}$ and $t > s_{i(x)}$. Stage r^* is a σ-stage, so $r^* = s_j$ for some $j \geqslant i(x)$, and therefore $x < \text{dom } \Phi_e(A, B)[r^*] = m^\tau(p)$.

(ii) Since p is permitted at t, we must have dom $\Phi_e(A, B)[t] \geqslant m^\tau(p) > x$, so $\varphi_{e,t}(x)$ is defined. Suppose $A_t \oplus B_t \upharpoonright \varphi_{e,t}(x) \neq A_s \oplus B_s \upharpoonright \varphi_{e,t}(x)$; if this was caused by a pair $\langle \nu, z \rangle$ with ν stronger than σ, then p would have been cancelled in the interval $[t, s)$. If ν is weaker than σ, then ν was initialised at t, and z must have been appointed (and later permitted, if ν works for a P requirement) at some stage $r \in (t, s)$. If ν works for an R requirement, $z > \varphi_{e,t}(x)$; if ν works for a P requirement, we lift the use of z to be large when it is permitted. In either case, ν cannot injure the computation $\Phi_e(A, B, x)[t]$.
Hence the injury must have been caused by $\langle \sigma, p \rangle$; but at stage t we set $\lambda_{d,t+1}(p) > \varphi_{e,t}(x)$, so it is too large to injure A below $\varphi_{e,t}(x)$. Hence $A_t \oplus B_t \upharpoonright \varphi_{e,t}(x) = A_s \oplus B_s \upharpoonright \varphi_{e,t}(x)$.

(iii) Since $A_t \oplus B_t \upharpoonright \varphi_{e,t}(x) = A_s \oplus B_s \upharpoonright \varphi_{e,t}(x)$, we must have $\varphi_{e,s}(x) = \varphi_{e,t}(x)$. Then $\lambda_{d,s}(p) \geqslant \lambda_{d,t+1}(p) > \varphi_{e,t}(x) = \varphi_{e,s}(x)$.

□

LEMMA 2.4.5: *Let τ be a node working for Q_e. Let $\rho \succeq \tau\hat{\ }\infty$ be a node working for R_d, and let y be a follower of ρ such that y is appointed by stage $s \geqslant s_{i(x)}$. Suppose that $\langle \rho, y \rangle \notin b(x)$. Then:*

(i) Let t be the stage at which y was appointed. Then $m^\tau(y) > x$ and $x \in$ dom $\Phi_e(A, B)[t]$;

(ii) $A_t \oplus B_t \upharpoonright \varphi_{e,t}(x) = A_s \oplus B_s \upharpoonright \varphi_{e,t}(x)$;

(iii) $y > \varphi_{e,s}(x)$.

Proof. (i) Since y is appointed at t it must be a ρ-stage, and since we have $\langle \rho, y\rangle \notin b(x)$, $t \geqslant s_{i(x)}$; thus t must be s_j for some $j \geqslant i(x)$. Therefore $x < \text{dom } \Phi_e(A, B)[t] = m^\tau(y)$.

(ii) By the same reasoning as Lemma 2.4.4 (ii), if $A_t \oplus B_t \restriction \varphi_{e,t}(x) \neq A_s \oplus B_s \restriction \varphi_{e,t}(x)$ then this is caused by $\langle \rho, y\rangle$ itself. However, y is appointed to be large at t, so certainly $y > \varphi_{e,t}(x)$.

(iii) Since $A_t \oplus B_t \restriction \varphi_{e,t}(x) = A_s \oplus B_s \restriction \varphi_{e,t}(x)$, we have $\varphi_{e,t}(x) = \varphi_{e,s}(x)$; hence $y > \varphi_{e,s}(x)$.

□

Lemma 2.4.6: *Let τ be a node on the true path working for Q_e. Then $\tau\hat{}\infty$ is on the true path if, and only if, $\Phi_e(A, B)$ is total.*

Proof. $\Rightarrow$ Suppose that $\tau\hat{}\infty$ is on the true path, and fix $x < \omega$. The sets $a(x)$ and $b(x)$ are both finite, and any pair $\langle \nu, z\rangle$ from either $a(x)$ or $b(x)$ can act and destroy the computation $\Phi_e(A, B, x)$ only finitely many times. Therefore, there are only finitely many $j \geqslant i(x)$ such that either $A_{s_j} \restriction \varphi_{e,s_j}(x) \neq A_{s_{j+1}} \restriction \varphi_{e,s_j}(x)$ or $B_{s_j} \restriction \varphi_{e,s_j}(x) \neq B_{s_{j+1}} \restriction \varphi_{e,s_j}(x)$, and hence $x \in \text{dom } \Phi_e(A, B)$.
$\Leftarrow$ Suppose that $\Phi_e(A, B)$ is total. Then there are infinitely many expansionary stages for $\Phi_e(A, B)$, and so infinitely many τ-stages will also be $\tau\hat{}\infty$-stages. Hence, $\tau\hat{}\infty$ is on the true path.

□

We now prove that the P requirements are met. This, of course, requires that Λ_e is total for every e; the proof of this is precisely that of Lemma 2.1.9.

Lemma 2.4.7: *For all $i, e < \omega$, the requirement P_e^i is met.*

Proof. Fix i, e, and let σ be the node on the true path working for P_e^i. Let s^* be the last stage at which σ is initialised. There are then three possibilities for σ:

(i) σ has a follower $p = p(\sigma, j)$ that is never realised;

(ii) All followers of σ are realised but none is permitted;

(iii) A follower of σ is permitted.

If case (i) holds, then $p \notin \text{dom } o^i$, and P^i_e is met by false hypothesis.

Suppose then that case (ii) holds. Then σ appoints an infinite sequence of followers after stage s^*. Suppose $p = p(\sigma, j)$ is appointed at some stage $s_0 > s^*$. Let t be the stage at which p becomes realised; at t, we define $m^\tau(p)$ for all $\tau \in \text{prec}(\sigma)$. For each such τ, we have $\tau\hat{}\infty$ on the true path. By Lemma 2.4.6, there is a stage s at which for all $\tau \in \text{prec}(\sigma)$, for all $x < m^\tau(p)$, $x \in \text{dom } \Phi_e(A, B)[s]$ by an A-, B-correct computation. Given A and B we can find such an s. Then $W_{e,s} \upharpoonright j + 1 = W_e \upharpoonright j + 1$, as for all $k < j$, if $k \notin W_{e,s}$ then $p(\sigma, k)$ was appointed and realised before p was. Hence, for all $k \leqslant j$, for all $\tau \in \text{prec}(\sigma)$, $m^\tau(p(\sigma, k)) \leqslant m^\tau(p)$. If $k \leqslant j$ is such that $k \notin W_{e,s}$ but $k \in W_{e,s'}$ for some $s' > s$, then $p(\sigma, k)$ would be permitted at s'.

Finally, suppose case (iii) holds, and let $p = p(\sigma, j)$ be the permitted follower. By Lemma 2.4.2, σ must only act finitely often for p, so let t be the least stage such that σ never acts for p after stage t.
At stage t we define $\Lambda_e(A, W_e, p)[t + 1] = t + 1 \neq f^i_t(p)$. Suppose that $f^i(p) = \Lambda_e(A, W_e, p)[t + 1]$. Then $f^i(p) \neq f^i_t(p)$, so there is a stage $s > t$ such that for all $r \geqslant s$ we see $f^i_r(p) = f^i(p)$. Since σ is on the true path, it must become accessible again at some stage $r \geqslant s$; at this stage, we redefine $\Lambda_e(A, W_e, p)[r] \neq f^i(p)$.

$\square$

Finally, we ensure that there is an ω-computable approximation for every $\Phi_e(A, B)$ which is total, and thus prove that the Q requirements are satisfied.

Lemma 2.4.8: *For all $e < \omega$, the requirement Q_e is met.*

Proof. Suppose that $\Phi_e(A,B)$ is total, and τ is the node on the true path working for Q_e. We proceed to build functions g, m such that $\langle g_s, m_s\rangle_{s<\omega}$ is an ω-computable approximation for $\Phi_e(A,B)$. Let $s^*, \langle s_i\rangle_{i<\omega}, i(x), a(x), b(x)$ be as defined prior to Lemma 2.4.4.

For all $x<\omega$, and all $j \geqslant i(x)$, we define $g_j(x) = \Phi_e(A,B,x)[s_j]$.

Fix $x<\omega$, and let $j \geqslant i(x)$. Let $\langle\sigma_1,p_1\rangle, \langle\sigma_2,p_2\rangle, \ldots, \langle\sigma_n,p_n\rangle$ be the members of $a_j(x)$. For each $1 \leqslant k \leqslant n$, suppose σ_k works for requirement $P^{i_k}_{d_k}$, and let $t_{k,j}$ be the last stage prior to s_j at which σ_k acted for p_k, including granting permission. If there is no such stage, let $t_{k,j}$ be the stage at which p_k was appointed.
We then define $m_j(x) = o^{i_1}_{t_{1,j}}(p_1) + o^{i_2}_{t_{2,j}}(p_2) + \ldots + o^{i_n}_{t_{n,j}}(p_n) + |b_j(x)|$.

It is clear that $\lim_s g_s(x) = \Phi_e(A,B,x)$, and that m is non-increasing. We then require that if $g_j(x) \neq g_{j+1}(x)$, we see $m_j(x) > m_{j+1}(x)$. Since all the summands of $m_j(x)$ are smaller than ω and as such are natural numbers, if there is a decrease in any one summand, then the entire sum will decrease. The value of $m_j(x)$ is also bounded by ω.

If $g_j(x) \neq g_{j+1}(x)$ due to change in A, then there is a pair $\langle\sigma,p\rangle \in a_j(x)$ such that σ working for P^i_d enumerates into A at stage s_j. Hence we must have $\Lambda_d(A,W_d,p)[s_j] = f^i_{s_j}(p)$.
Let k be such that $\sigma=\sigma_k$. At stage $t_{k,j}$, we defined $\Lambda_d(A,W_d,p)[t_{k,j}+1] = t_{k,j}+1 > f^i_{t_{k,j}}(p)$. Hence from $f^i_{s_j}(p) = \Lambda_d(A,W_d,p)[s_j]$ we may conclude that $f^i_{t_{k,j}}(p) \neq f^i_{s_j}(p)$, and hence $o^i_{t_{k,j}}(p) > o^i_{s_j}(p)$. Given that $t_{k,j+1} = s_j$, we then have $o^i_{t_{k,j}}(p) > o^i_{t_{k,j+1}}(p)$.

If $g_j(x) \neq g_{j+1}(x)$ due to change in B, then it is caused by a pair $\langle\rho,y\rangle \in b_j(x)$ enumerating y into B_{s_j+1}. But then $y \in B_{s_{j+1}}$, so $\langle\rho,y\rangle \notin b_{j+1}(x)$, and hence $|b_j(x)| > |b_{j+1}(x)|$.

Therefore, if $g_j(x) \neq g_{j+1}(x)$, we see a decrease in at least one summand of $m_j(x)$, and consequently $m_j(x) > m_{j+1}(x)$. Hence, $\langle g_s, m_s \rangle_{s<\omega}$ is an ω-computable approximation of $\Phi_e(A, B)$, and Q_e is met.

□

This concludes the proof of Theorem 2.4.1. The theorem that follows presents a minor modification on the construction of Theorem 2.4.1; so minor, that we omit the verification as it would simply be repetition. However, instead of building a maximal totally ω-c.a. c.e. degree, we build a chain of totally ω-c.a. c.e. degrees which is cofinal in the cone of totally ω-c.a. c.e. degrees above the chain's least member. That is, all totally ω-c.a. c.e. degrees which bound the minimum degree in the chain are themselves bounded by a member of the chain. This theorem holds for any $\alpha \leqslant \varepsilon_0$ in place of ω.

THEOREM 2.4.9: *There is a totally ω-c.a. c.e. degree $\boldsymbol{a}$ such that, for any c.e. degree $\boldsymbol{b}$, if $\boldsymbol{b} \geqslant \boldsymbol{a}$ then $\boldsymbol{b}$ is not maximal totally ω-c.a..*

We build an infinite chain of c.e. sets $(A =)B_0 <_T B_1 <_T B_2 <_T \cdots$, by building c.e. sets $A, D_1, D_2, \ldots$ and setting $B_k = B_0 \oplus D_1 \oplus \cdots \oplus D_k$. We require that, for all i, $\deg_{\mathrm{T}}(B_i)$ is totally ω-c.a., and if $\deg_{\mathrm{T}}(A \oplus W_e)$ is totally ω-c.a. then $A \oplus W_e \leqslant_T B_k$ for some k. If we can successfully build such an A, then $\deg_{\mathrm{T}}(A) = \boldsymbol{a}$.

Requirements:

Let $\langle \Psi_e \rangle_{e<\omega}, \langle \Phi_e \rangle_{e<\omega}$ each be an enumeration of all consistent functionals. It is clear that for all i, we have $B_i \leqslant_T B_{i+1}$. To ensure that we also have $B_i \not\geqslant_T B_{i+1}$, we need to meet the set of requirements given by

$$\text{for all } i, e < \omega,\ R^i_e : \Phi_e(B_i) \neq D_{i+1}.$$

We require that each B_i is totally ω-c.a., and achieve this by meeting

$$\text{for all } i, e < \omega,\ Q_e^i : \text{If } \Psi_e(B_i) \text{ is total, then it is } \omega\text{-c.a..}$$

Finally, we require the crucial property that every totally ω-c.a. degree above $\boldsymbol{a}$ is below $\deg_{\mathrm{T}}(B_k)$ for some k. We enumerate functionals Λ, Δ, to meet the requirements

$$\text{for all } i, e < \omega,\ P_e^i : \text{If } \bigcup_s \operatorname{dom} o_s^i = \omega, \text{ then either } \Lambda_e(A, W_e) \neq f^i,$$
$$\text{or } \Delta_e^i(A, D_1, \ldots, D_k) = W_e,$$

where $D_1, \ldots, D_k$ are particular to $\langle i, e \rangle$, and are those D that appear in requirements stronger than P_e^i.

Discussion:

The construction functions in much the same way as that of Theorem 2.4.1. As in that construction, nodes working for P or R requirements are capable of injuring Q requirements by destroying computations with enumeration into A or some D_i. We deal with this in precisely the same manner: for every follower, we define m^τ as an indicator of which computations the follower may not injure, and proceed to ensure that none of these computations suffer injury (from the follower) as the strategy progresses. In fact, the only way in which the strategy tree here truly differs from that of Theorem 2.4.1 is that, at σ working for P_e^i, we build the functional Δ_σ. (Even this closely follows the building of the functional of the same name in the proof of Theorem 2.1.3.) The entire thing proceeds rather pleasantly, and without complication. As such, the verification is the same as that of Theorem 2.4.1 with only differing notation, except for Lemma 2.4.7. To prove that $\Delta_e^i(A, D_1, \ldots, D_k) = W_e$ when we cannot ensure $\Lambda_e(A, W_e) \neq f^i$, we insert the argument used in Lemma 2.1.10 that addresses the Δ functional built for Theorem 2.1.3.

2.5. Uniformly Totally α-c.a. Degrees

Let $\alpha \leqslant \varepsilon_0$. We call $h : \omega \to \alpha$ an *α-order function* if h is nondecreasing, computable, and its range is unbounded in α.

Let $\langle f_s, o_s\rangle_{s<\omega}$ be an α-computable approximation. We then call $\langle f_, o_s\rangle_{s<\omega}$ an h-*computable approximation* if, for all x, we have $o_0(x) \leqslant h(x)$. Just as for the $\mathscr{R}$-c.a. functions, for any α-order function h we can produce a list that exhausts all h-c.a. functions:

LEMMA 2.5.1: *Let $\alpha \leqslant \varepsilon_0$ and let h be an α-order function. Then there is an effective enumeration $\langle f^e_s, o^e_s\rangle$ of total $(h+1)$-computable approximations such that letting $f^e = \lim_s f^e_s$, the sequence $\langle f^e\rangle$ contains all h-c.a. functions.*

For every ordinal $\alpha \leqslant \varepsilon_0$, there exists an α-order function. Therefore, a function f is α-c.a. if, and only if, it is h-c.a. for some α-order function h: any h-c.a. function is clearly α-c.a., and if f has an α-computable approximation $\langle f_s, o_s\rangle$, define $\hat{h}(x) = \max\{h(x), o_0(x)\}$. Then $\hat{h}$ is an α-order function, and f is $\hat{h}$-c.a..

Let $\boldsymbol{d}$ be a Turing degree. As it happens, every function $f \in \boldsymbol{d}$ is h-c.a. for some α-order function h if, and only if, every $f \in \boldsymbol{d}$ is h-c.a. for *every* α-order function h. We use this to produce a subclass of the α-c.a. degrees, namely the *uniformly totally α-c.a.* degrees. For $\alpha \leqslant \varepsilon_0$, we say that a Turing degree $\boldsymbol{d}$ is uniformly totally α-c.a. if for some (all) α-order function h, every $f \in \boldsymbol{d}$ has an h-computable approximation.
Equivalently, a c.e. Turing degree is uniformly totally α-c.a. if, and only if:

(i) for some (all) α-order function h, every $f \leqslant_{\mathrm{T}} \boldsymbol{d}$ is h-c.a.; and,

(ii) for some (all) α-order function f, every set in $\boldsymbol{d}$ is h-c.a..

Let $\alpha < \varepsilon_0$, and let $h : \omega \to \alpha + 1$ be the constant function with value α. Then every totally α-c.a. degree is also uniformly totally $(\alpha+1)$-c.a., as witnessed by $(\alpha+1)$-order function h.

Let β be an ordinal which is not a power of ω, so for some ordinal γ we have $\beta \in (\omega^\gamma, \omega^{\gamma+1})$. By Theorem 1.4.1, any uniformly totally β-c.a. degree is also totally ω^γ-c.a.. Thus, for such β, there is an ordinal α which is a power of ω such that the set of uniformly totally β-c.a. degrees is exactly the set of totally α-c.a. degrees. The uniformly totally β-c.a. degrees might only then be distinguished from the hierarchy of totally α-c.a. degrees when β is a power of ω. As the following theorem from [4] shows, they do in fact form a distinct level of the hierarchy in this case.

THEOREM 2.5.2: *Let $\alpha \leqslant \varepsilon_0$ be a power of ω.*

(1) There is a uniformly totally α-c.a. c.e. degree which is not totally γ-c.a. for any $\gamma < \alpha$.

(2) There is a totally α-c.a. c.e. degree which is not uniformly totally α-c.a..

The last theorem in this thesis examines the cone below a c.e. degree $\boldsymbol{d}$ which is *not* totally α-c.a. for $\alpha \leqslant \varepsilon_0$. The construction builds a degree bounded by $\boldsymbol{d}$ which is totally α-c.a. but not uniformly so. We use non-totally α-c.a. permitting (from mind-changes in a computable approximation $\langle f_s \rangle$ of a function $f \in \boldsymbol{d}$ which is not α-c.a.) to allow nodes to enumerate; we do this to ensure that the set built by us is in fact computable from $\boldsymbol{d}$.

THEOREM 2.5.3: *Let $\alpha \leqslant \varepsilon_0$ be a power of ω, and let $\mathbf{d}$ be a c.e. Turing degree which is not totally α-c.a.. There is a degree $\boldsymbol{a} < \mathbf{d}$ such that $\boldsymbol{a}$ is totally α-c.a., but not uniformly totally α-c.a..*

Let $\langle D_s \rangle_{s<\omega}$ be a given computable enumeration of $D \in \boldsymbol{d}$, and let Ψ be a given, fixed functional such that $\Psi(D)$ has no α-computable approximation. By accelerating the enumeration of D, we may assume that for all s, for all $x < s$, we have $\Psi(D, x) \downarrow [s]$. We proceed to build a c.e. set A with the intent that $\deg_{\mathrm{T}}(A) = \boldsymbol{a}$.

Requirements:

Let $\langle \Phi_e \rangle_{e<\omega}$ be an enumeration of all consistent functionals. Firstly, we must build A to be totally α-c.a., by meeting the set of requirements given by:

$$\text{for all } e < \omega,\ Q_e : \text{If } \Phi_e(A) \text{ is total, then it is } \alpha\text{-c.a..}$$

To ensure that A is built to be not uniformly totally α-c.a., we fix an order function $h : \omega \to \alpha$. Let $h+1$ denote the function given by $n \mapsto h(n)+1$. By Lemma 2.5.1, there is an effective enumeration $\langle\langle f_s^i, o_s^i\rangle_{s<\omega}\rangle_{i<\omega}$ of total $(h+1)$-computable approximations such that letting $f^i = \lim_s f_s^i$, the sequence $\langle f^i \rangle_{i<\omega}$ contains all h-c.a. functions. We enumerate a functional Λ, and with it meet the following requirements:

$$\text{for all } i < \omega,\ P^i : \Lambda(A) \neq f^i.$$

Globally, we also ensure that $A \leqslant_{\mathrm{T}} D$.

Discussion:

To meet P^i in isolation, we choose a follower p and wait until such time as we see $\Lambda(A,p)[s] = f_s^i(p)$. If we see this, we enumerate $\lambda_s(p)$ into A_{s+1} and force a disagreement. As $f^i(p)$ may change up to $o_0(p)$-many times (where $o_0(p) \leqslant h(p)+1 < \alpha$), we will need to enumerate $\lambda(p)$ into A at most $o_0(p)$-many times.

Of course, this enumeration can destroy a computation $\Phi_e(A,x)$, and be thereby problematic to a bound appointed to x for Q_e. To avoid this, we prevent p from injuring any computation already in place at the stage at which p is appointed. When we first see $\Phi_e(A,x)\downarrow$, there must then be only a finite number of followers capable of injuring the computation. Let $p_1,\ldots,p_n$ be these followers, and let s be the stage; each p_j will act at most $o_s(p_j)$-many times. Since α is a power of ω, the sum $o_s(p_1)+\cdots+o_s(p_n)$ is less than α, and is a bound on the number of times the computation $\Phi_e(A,x)$ will be destroyed.

To ensure $A \leqslant_{\mathrm{T}} D$, we require that every enumeration into A be associated with an enumeration into D. To each follower p of a node σ working for a P^i requirement, we assign a number $j < \omega$ and set $\delta(p) = \psi(j)$; we say that *p monitors input j*. If we then see change in D below $\delta(p)$, we afford p the opportunity to act at that stage. If p does act (i.e. if $\Lambda(p) = f^i(p)$ at that stage), we initialise weaker nodes and cancel any larger followers in place at the same node. These larger followers will each have been monitoring at least one input; any such input must then become monitored by p. We then return to observing for change in D, but now set $\delta(p)$ to be the largest ψ-use of all inputs monitored by p. We note that p will always monitor an interval of ω, and all followers of σ will collectively monitor an initial segment of ω. Suppose p is a follower of σ working for P^i. We cannot be sure that there will be sufficient change in D below $\delta(p)$ to allow p to act enough times to force $\Lambda(A, p) \neq f^i(p)$. At each σ-stage s, we check whether we appear to be in danger of not meeting P^i: if every follower p of σ is such that $\Lambda(A, p)[s] = f^i_s(p)$, we appoint an additional follower to monitor the least j currently unmonitored at σ. We must initialise weaker nodes than σ when we appoint a new follower, to prevent intolerable injury to Q requirements. In verifying our construction, we argue that there must be a follower of σ which receives sufficiently many permissions, otherwise $\Psi(D)$ is α-computably approximable.

Strategy Tree Construction:

A node τ working for Q_e has two outcomes, $\infty <$ f, which measure whether $\Phi_e(A)$ is expansionary at any stage where τ is accessible. A node σ working for P^i has a unique outcome.

We order the requirements in order type ω. Let s be a stage.

First at s, we check for followers ready to permit and enumerate. We do this by checking for a follower p at a node σ working for P^i such that:

(i) a number less than $\delta_s(p)$ enters D_s; and,

(ii) $\Lambda(A,p)[s] = f^i_s(p)$.

If there is such a follower, we select the strongest i.e. that which is appointed to the strongest node, and is the least follower of that node for which (i) and (ii) hold. For this p, we enumerate $\lambda_s(p)$ into A_{s+1}. For every follower p' of σ such that $p' > p$, we cancel p' and let p take over those inputs of $\Psi(D)$ monitored by p'. We then redefine $\Lambda(A,p)[s+1] = s+1$ with large use, initialise nodes weaker than σ, and end the stage.

If no follower is permitted, we proceed to build the path γ_s of accessible nodes.

Suppose $\tau \in \gamma_s$ works for Q_e. Let $t < s$ be the last stage before s at which $\tau\hat{}\infty$ was accessible, or $t = 0$ if there is no such stage. If dom $\Phi_e(A)[s] \geqslant t$, let $\tau\hat{}\infty \in \gamma_s$; otherwise, let $\tau\hat{}\mathrm{f} \in \gamma_s$.

Suppose $\sigma \in \gamma_s$ works for P^i. If σ has at least one follower, and there is a follower p of σ such that $\Lambda(A,p)[s] \neq f^i_s(p)$, perform no action at σ this stage. Let the successor node of σ be accessible.
Otherwise, let j be the least input of $\Psi(D)$ not monitored by a follower of σ. We appoint a new, large follower p, and set it to initially monitor j. We then define $\Lambda(A,p)[s+1] = s+1$ with use p, initialise all weaker nodes, and end the stage.

At the conclusion of stage s, we maintain the functional Λ and update δ for every follower. For any $p \leqslant s$ for which $\lambda(p)$ was not already redefined at s, if $p \notin$ dom $\Lambda_s(A_{s+1})$ we let $\Lambda(A,p)[s+1] = s+1$ with use $\lambda_{s+1}(p) = 0$. For any follower $p(\sigma, s)$ which is still in place at stage $s+1$, we define $\delta_{s+1}(p) = \max\{\psi_{s+1}(j) \mid j \text{ monitored by } p \text{ at stage } s+1\}$. By our assumption that $\Psi(D,x)\downarrow [s+1]$ for all $x < s+1$, this value will exist.

Verification:

We proceed to take up the usual gauntlet of claims that require verification: that Λ is total, that the true path is infinite, that the nodes on the true path ensure the requirements are met.

LEMMA 2.5.4: *Λ is total.*

Proof. Let $p < \omega$, and suppose that $p \notin \text{ dom } \Lambda(A)$.
If p is never appointed as a follower to some σ, or is appointed and later cancelled, then at the least stage $s > p$ at which p is not a follower we define $\Lambda(A,p)[s+1] = s+1$ with use 0.
Otherwise, p is a follower for some σ which is appointed and never cancelled. Suppose $\Lambda(A,p)$ is destroyed and redefined infinitely often. This destruction can be caused by p itself only finitely many times; hence, it must be enumeration by other followers that destroys $\Lambda(A,p)$ infinitely often. If $\Lambda(A,p)$ is instead destroyed and never redefined, this must have also been due to the action of another follower.
Suppose follower p' of σ' enumerates $\lambda_s(p)$ into A_{s+1}. If p' was appointed after p, then $\lambda_s(p)$ is too large to injure $\Lambda(A,p)[s]$. Hence, p' must have been appointed prior to p, and since p' is not cancelled when p is appointed, σ' must be stronger than σ. Then the enumeration by p' at stage s cancels p; in maintaining Λ, we then define $\Lambda(A,p)[s+1] = s+1$ with use 0 at the earliest stage $s' \geqslant p$. Hence, any action of another follower that destroys $\Lambda(A,p)$ is certainly followed by a redefinition that is permanent, so $p \in \text{dom } \Lambda(A)$.

$\square$

To prove that the P requirements are met, we must use the fact that $\boldsymbol{d}$ is not totally α-c.a.; specifically, that $\Psi(D)$ is not an α-c.a. function. To meet its requirement, a node σ on the true path working for P^i must (eventually) have a follower p for which we are able to force the disagreement $\Lambda(A,p) \neq f^i(p)$. We only allow σ to enumerate into A when there is change below $\delta(p)$, i.e. below the maximum use $\psi(j)$ of the inputs it monitors. It is entirely possible that any particular follower p of σ will not see enough changes below $\delta(p)$ to allow σ to permit p the required number of times. What we need, and now proceed to show, is that at least one follower *must* receive enough permissions to meet the requirement; if none does, then we can use that to build an α-computable approximation for $\Psi(D)$.

LEMMA 2.5.5: *The construction is fair to nodes on the true path.*

LEMMA 2.5.6: *For all $i < \omega$, P^i is met. In particular, let σ be a node on the true path working for P^i: if there is no permanent follower p of σ that receives sufficiently many permissions to ensure $\Lambda(A, p) \neq f^i(p)$, then $\Psi(D)$ is α-computably approximable.*

We prove Lemmata 2.5.5 and 2.5.6 simultaneously.

Proof. Suppose Lemma 2.5.5 to be true up to $\mu \prec \delta_\omega$. The node μ will only initialise weaker nodes if it works for a requirement P^i, in which case it will initialise and end the stage whenever, for every follower p of μ, we see $\Lambda(A, p)[s] = f^i_s(p)$, or when a follower of μ is permitted and enumerates into A. Any one follower of μ may be permitted only finitely many times; hence, if this happens infinitely often, then the construction has not been able to force a difference at a permanent follower of μ.

We now assume that there is no permanent follower p of μ that receives sufficiently many permissions to ensure $\Lambda(A, p) \neq f^i(p)$, and show that $\Psi(D)$ is consequently α-computably approximable. From this assumption, it follows that infinitely-many followers are appointed so that each $j < \omega$ is eventually monitored by a follower of μ, and that for each follower p of μ at stage s, if p is not ever cancelled then there is a stage $t \geqslant s$ at which $\Lambda(A, p)[t] = f^i_t(p)$.

Let s^* be the last stage at which μ is initialised. We proceed to build $\langle g_s, m_s \rangle_{s<\omega}$, an α-computable approximation of $\Psi(D)$.

Let $j < \omega$ be fixed, let $s > s^*$ be a stage. Let $p = p_{j,s}$ be the follower monitoring j at s; if there is no such follower, then we leave $g_s(j)$ and $m_s(j)$

undefined. Suppose then that there is such a follower p. If $\Lambda(A,p)[s] = f^i_s(p)$, we let $g_s(j) = \Psi(D,j)[s]$. Otherwise, let $t < s$ be the last stage at which j had a defined monitor $p_{j,t}$ and $\Lambda(A,p_{j,t})[t] = f^i_t(p_{j,t})$. If such a stage exists, define $g_s(j) = g_t(j)$; otherwise, leave $g_s(j)$ and $m_s(j)$ undefined.
Let $q_0, q_1, \ldots, q_k$ be the followers of μ less than p at stage s, ordered by increasing size. We note that if p is cancelled at some future stage it is by permission of one of these q, and that q will take over the monitoring of j.
If $\Lambda(A,p)[s] = f^i_s(p)$, define $m_s(j) = o^i_s(q_0) + \cdots + o^i_s(q_k) + o^i_s(p)$. Otherwise, define $m_s(j) = m_t(j)$ for t as previously described.

Suppose that $g_s(j) \neq g_{s+1}(j)$ for some s. Let $t \leqslant s$ be the last stage at which $\Lambda(A,p_{j,t})[t] = f^i_t(p_{j,t})$, so $g_s(j) = g_t(j)$ and $m_s(j) = m_t(j)$. If $p_{j,t} \neq p_{j,s+1}$, then a permission has caused a smaller follower q to take over the monitoring of j. We set $\Lambda(A,q)[t+1] = t+1 \neq f^i_t(q)$, but $\Lambda(A,q)[s+1] = f^i_{s+1}(q)$; therefore, $f^i_t(q) \neq f^i_{s+1}(q)$, so $o^i_t(q) \neq o^i_{s+1}(q)$. All larger followers than q are cancelled at t, so $o^i_{s+1}(q)$ is the final summand of $m_{s+1}(j)$. Hence, $m_s(j) > m_{s+1}(j)$.
Assume then that $p_{j,t} = p_{j,s+1} = p$. If $\Lambda(A,p)[t+1] \neq f^i_{t+1}(p)$ because the value of $f(p)$ changed, then certainly $o^i_t(p) > o^i_{t+1}(p) \geqslant o^i_{s+1}(p)$, so $m_t(j) = m_s(j) > m_{s+1}(j)$. Otherwise, we cause the change by redefining $\Lambda(A,p)[t+1] = t+1 \neq f^i_t(p)$. Then $f^i_t(p) \neq f^i_{s+1}(p)$ to the same effect as when $f(p)$ alone changes: specifically, $m_t(j) = m_s(j) > m_{s+1}(j)$.

Since each term $o^i_s(p)$ is bounded by α and α is a power of ω, the sum $m_s(j)$ is bounded by α. The function m also inherits the non-increasing nature of its constituent parts.
Suppose then that $\lim_s g_s(j) \neq \Psi(D,j)$. Let t be least such that for all stages $t' \geqslant t$, we have $g_{t'}(j) = \lim_s g_s(j)$. Let $p = p_{j,t}$; then $\Lambda(A,p)[t] = f^i_t(p)$. Since $\Psi(D,j)[t] \neq \Psi(D,j)$, there must be some least stage $r \geqslant t$ such that we see some value $x < \psi_t(j)$ enter $D_r + 1$. Then some follower $q \leqslant p$ of μ must be permitted at stage r (p itself is certainly eligible), ensuring

$\Lambda(A, q)[r+1] \neq f_r^i(q)$ for $q = p_{j,r+i}$. Assuming q itself is not cancelled by another permission, there is a stage $r' > r$ at which $\Lambda(A, q)[r'] = f_{r'}^i(q)$; at this stage, we redefine $g_{r'}(j) = \Psi(D, j)[r'] \neq g_t(j)$ - a contradiction. Hence $\lim_s g_s(j) = \Psi(D, j)$, and $\langle g_s, m_s \rangle_{s<\omega}$ is therefore an α-computable approximation of $\Psi(D)$. Since we know $\Psi(D)$ to have no such approximation, our hypothesis that no follower receives enough permissions to force $\Lambda(A) \neq f^i$ must be false. Hence, μ is eventually appointed a follower p for which, for some stage $s > s^*$ and for all stages $t > s$, we have $\Lambda(A, p)[t] \neq f_t^i(p)$; after stage s, μ will not ever initialise weaker nodes.

□

COROLLARY 2.5.7: *The true path is infinite.*

Proof. Given Lemma 2.5.5, the true path is infinite if we have infinitely many stages at which nodes are accessible. Suppose instead that cofinitely many stages are permission stages; let s be the last stage at which δ_s is defined (non-empty). The finitely-many followers in place at stage $s+1$ must then receive infinitely many permissions. In particular, one follower p for a node σ working for P^i must receive infinitely many permissions at stages $s_0 < s_1 < \cdots$ after stage s. Then $o_{s_0}^i(p) > o_{s_1}^i(p) > \cdots$ is an infinite descending sequence of ordinals - a contradiction.

□

Hence, the true path contains a node for each P and Q requirement; further, those that work for P requirements are successful in satisfying their associated requirement. We now verify that the Q requirements are met, in the manner which by now must be very familiar to the reader.

LEMMA 2.5.8: *Let σ be a node working for P^i, let p be a follower of σ appointed at stage s, and suppose $\Phi_e(A, x)\downarrow[s]$. If the computation $\Phi_e(A, x)[s]$ is destroyed at some later stage $t > s$, then p is cancelled at t.*

Proof. If $\Phi_e(A, d)[s]$ is destroyed at t, it is due to enumeration into A by some node σ' working for $P^{i'}$. If σ' is stronger than σ, then σ is initialised

at t and p is thereby cancelled.
Otherwise, σ' is weaker than, or is, σ. Let p' be the particular follower that σ' acts for at t; since followers are appointed large and p' is smaller than $\varphi_{e,s}(x)$, p' must have been appointed prior to stage s. If σ' is weaker than σ, then the appointment of p at s initialised σ', cancelling p' and preventing it from injuring $\Phi_e(A,x)[s]$. If σ' is σ then, since $p' < p$, the permission of p' at stage t cancels p.

$\square$

Let $\tau\hat{}\infty$ be on the true path, where τ is a node working for requirement Q_e. Let s^* be the last stage at which $\tau\hat{}\infty$ is initialised, and let $s_0 < s_1 < \cdots$ be the stages after s^* at which $\tau\hat{}\infty$ is accessible. We fix $x < \omega$; define $i(x)$ to be the least j such that $x <$ dom $\Phi_e(A)[s_j]$.
We then define $a(x)$ to be the set of all pairs $\langle \sigma, p\rangle$ such that σ works for a requirement P^i, and p is a follower of σ appointed before, but not cancelled by, stage $s_{i(x)}$. By Lemma 2.5.8, any follower appointed at or later than stage $s_{i(x)}$ is incapable of injuring $\Phi_e(A,x)$; $a(x)$ thus contains all node/follower pairs capable of causing injury to the computation $\Phi_e(A,x)$.
Similarly, for each $j \geqslant i(x)$, we define $a_j(x)$ to be the set of all pairs $\langle \sigma, p\rangle$ such that $\langle \sigma, p\rangle \in a(x)$, and p is not cancelled by stage s_j.

Lemma 2.5.9: *For all $e < \omega$, the requirement Q_e is met.*

Proof. Fix $e < \omega$; if $\Phi_e(A)$ is total, let τ be the node on the true path that works for Q_e. Then $\tau\hat{}\infty$ is also on the true path. We proceed to define functions g, m, such that $\langle g_s, m_s\rangle_{s<\omega}$ is an α-computable approximation for $\Phi_e(A)$. Fix $x < \omega$, and let $a(x)$, $a_j(x)$ be as previously described.

For all $j \geqslant i(x)$, define $g_j(x) = \Phi(A,x)[s_j]$.

Fix $j \geqslant i(x)$. Let $\langle \sigma_0, p_0\rangle, \langle \sigma_1, p_1\rangle, \ldots, \langle \sigma_n, p_n\rangle$ be the members of $a_j(x)$, in order of decreasing strength. Suppose that for all $0 \leqslant k \leqslant n$, the node σ_k works for requirement P^{i_k}. For each k, we let $t_{k,j}$ be the last stage before s_j

at which σ_k acted for p_k. If there is no such stage, let $t_{k,j}$ be the stage at which p_k was appointed.
We can now define $m_j(x) = o^{i_0}_{t_{0,j}}(p_0) + o^{i_1}_{t_{1,j}}(p_1) + \cdots + o^{i_n}_{t_{n,j}}(p_n)$.

It is clear that $\lim_s g_s(x) = \Phi_e(A, x)$. For all k and j, we have $o^{i_k}_{t_{k,j}}(p_k) \leqslant h(p_k) + 1 < \alpha$; as α is a power of ω, the sum $m_j(x)$ is also bounded by α.

Suppose that $g_j(x) \neq g_{j+1}(x)$; then $\Phi(A, x)[s_j] \neq \Phi(A, x)[s_{j+1}]$, and this change has been caused by a pair $\langle \sigma, p\rangle$ in $a_j(x)$ acting at stage s_j. If $\sigma = \sigma_k$, we must have $\Lambda(A, p)[s_j] = f^{i_k}_{s_j}(p)$.
At stage $t_{k,j}$, we defined $\Lambda(A, p)[t_{k,j} + 1] = t_{k,j} + 1 > f^{i_k}_{t_{k,j}}(p)$. Then, since $f^{i_k}_{s_j}(p) = \Lambda(A, p)[s_j]$, we may conclude that $f^{i_k}_{t_{k,j}}(p) \neq f^{i_k}_{s_j}(p)$. Since σ acts for p at s_j, we have $t_{k,j+1} = s_j$, and therefore $o^{i_k}_{t_{k,j}}(p) > o^{i_k}_{t_{k,j+1}}(p)$. Since σ initialises all weaker nodes at stage s_j, $\langle \sigma, p\rangle$ is then the weakest member of $a_{s_{j+1}}$, and so $o^{i_k}_{t_{k,j+1}}(p)$ is the last summand of $m_{j+1}(x)$. Consequently, $m_j(x) > m_{j+1}(x)$.

□

Finally, it is crucial that we have successfully built A and D such that $\deg_{\mathrm{T}}(A) < \deg_{\mathrm{T}}(D)$.

LEMMA 2.5.10: *The global requirement, $A \leqslant_T D$, is met.*

Proof. Given D, for any given $j < \omega$, we can certainly find a stage s at which $D_s \upharpoonright \psi_s(j) = D \upharpoonright \psi_s(j)$. Let p be a follower in place at stage $t \geqslant s$. If, for all followers p' at t such that $p' \leqslant p$, p' monitors at most j, we have $A_t \upharpoonright p = A \upharpoonright p$.

□

This concludes the proof of Theorem 2.5.3.

Chapter 3

Concluding Remarks

In [4], Downey and Greenberg establish that the hierarchy of totally α-c.a. degrees and uniformly totally α-c.a. degrees only collapses above $\mathbf{0}$ between $\alpha \leqslant \varepsilon_0$ for which α is a power of ω. Further, for α a power of ω, the uniformly totally α-c.a. degrees form a proper subset of the totally α-c.a. degrees, thus forming a distinct level in the hierarchy.

In this thesis, we proved several new facts about the hierarchy pertaining to collapse. Firstly, we showed that above any totally α-c.a. c.e. degree $\boldsymbol{a}$ there is no collapse in levels $\beta \geqslant \alpha^\omega$, due to the existence of a maximal totally β-c.a. c.e. degree above $\boldsymbol{a}$. Due to the manner in which that construction responds to enumeration into sets given by the opponent, it allows infinite positive action by nodes on the true path - a technique which may find uses elsewhere.

We then proved that above any totally ω-c.a. c.e. degree $\boldsymbol{a}$, there is a c.e. degree $\boldsymbol{d}$ which is totally ω^3-c.a. but not totally ω-c.a.. With slight modification, this same proof shows that above any totally ω^n-c.a. c.e. degree, there is a c.e. degree which is totally ω^{n+2}-c.a. but not totally ω^n-c.a.. The construction lacks a permitting mechanism, and uses an initial guess of ω when a bound on action by a follower is not yet known; this prevents us from making $\boldsymbol{d}$ totally ω^2-c.a. instead. Were we to find a proof that produced a totally ω^2-c.a., but not totally ω-c.a., c.e. degree (or a maximal totally ω^2-c.a. c.e. degree, to the same effect) above $\boldsymbol{a}$, it would likely be able to be generalised. We might then be able to prove that above every totally ω^n-c.a. c.e. degree there is a c.e. degree which is totally ω^{n+1}-c.a. but not totally ω^n-c.a.. Combined with Theorem 2.1.3, this would rule out any further collapse (anywhere) except between the powers of ω in the hierarchy of totally α-c.a. c.e. degrees. This remains open, and difficulties encountered thus far suggest the proof would be non-uniform. We have, however, proved this specifically for the cones above superlow c.e. degrees. The proof relies heavily on the fact that the jump function of a superlow c.e. set has an ω-computable approximation.

We then proved some theorems which make no statement on collapse, but which show the existence of interesting features. We proved that there is a pair $\boldsymbol{a}, \boldsymbol{d}$ of degrees such that $\boldsymbol{d}$ is totally ω-c.a. and acts as a 'minimal cover' for $\boldsymbol{a}$. That is, every totally ω-c.a. degree $\hat{\boldsymbol{a}} \geqslant \boldsymbol{a}$ is bounded by $\boldsymbol{d}$. We then extend this to produce an infinite chain of minimal cover-like degrees, so that every totally ω-c.a. degree which bounds the least member of the chain is itself bounded by a member of the chain. The degree $\boldsymbol{d}$ in the first of these theorems is necessarily maximal totally ω-c.a., yet in the extension, no member of the chain is, or is bounded by, a maximal totally ω-c.a. degree. Both of these results bear generalisation to any $\alpha \leqslant \varepsilon_0$ in place of ω. Finally, we proved that any c.e. degree which is not totally α-c.a. bounds a degree $\boldsymbol{a}$ which is totally, but not uniformly totally, α-c.a.. Then $\boldsymbol{a}$ is not totally γ-c.a. for any $\gamma < \alpha$, and as such itself bounds a totally, not uniformly totally, γ-c.a. degree.

These new results contribute to our understanding of this natural hierarchy as it pertains to the c.e. degrees, and are part of ongoing work.

Bibliography

[1] Cantor, G. Beitrage zur Begrundung der transfiniten Mengenlehre. *Math. Ann.*, 49 (1897), 207–246.

[2] Cholak, P., Coles, R., Downey, R., and Herrmann, E. Automorphisms of the lattice of $\prod_1^0$ classes: perfect thin classes and anc degrees. *Trans. Amer. Math. Soc*, 4899–4924.

[3] Cholak, P., Downey, R., and Walk, S. Maximal contiguous degrees. *J. Symbolic Logic 67*, 1 (2002), 409–437.

[4] Downey, R., and Greenberg, N. *A transfinite hierarchy of lowness notions in the computably enumerable degrees, unifying classes, and natural definability.* submitted.

[5] Downey, R., Greenberg, N., and Weber, R. Totally ω-computably enumerable degrees and bounding critical triples. *J. Math. Log. 7*, 2 (2007), 145–171.

[6] Downey, R., Jockusch, C., and Stob, M. *Recursion Theory Week: Proceedings of a Conference held in Oberwolfach, FRG, March 19–25, 1989.* Springer Berlin Heidelberg, Berlin, Heidelberg, 1990, ch. Array nonrecursive sets and multiple permitting arguments, pp. 141–173.

[7] Downey, R., Jockusch, C. G., and Stob, M. Array nonrecursive degrees and genericity. In *London Mathematical Society Lecture Notes Series 224* (1996), University Press, pp. 93–105.

[8] Ishmukhametov, S. Weak recursive degrees and a problem of spector. In *Recursion Theory and Complexity*, M. Arslanov and S. Lempp, Eds., vol. 2. de Gruyter, Berlin, 1999, pp. 81–87.

[9] Kummer, M. Kolmogorov complexity and instance complexity of recursively enumerable sets. *SIAM J. Comput. 25*, 6 (1996), 1123–1143.

[10] Schaeffer, B. Dynamic notions of genericity and array noncomputability. *Annals of Pure and Applied Logic 95*, 1-3 (1998), 37–69.

[11] Shoenfield, J. R. On degrees of unsolvability. *Ann. of Math. 69*, 2 (1959), 644–653.