TOTALLY ω-COMPUTABLY ENUMERABLE DEGREES I: BOUNDING CRITICAL TRIPLES

ROD DOWNEY, NOAM GREENBERG, AND REBECCA WEBER

Abstract. We characterize the class of c.e. degrees that bound a critical triple (equivalently, a weak critical triple) as those degrees that compute a function that has no ω-c.e. approximation.

1. Introduction

This paper is concerned with the long term programme which attempts to relate computational complexity as measured by, say, Turing reducibility, with definability within the structures of computability theory.

In particular, we will be concerned with definability issues within the computably enumerable Turing degrees. In terms of abstract results on definability, there has been significant success in recent years, culminating in Nies, Shore, Slaman [NSS98], where the following is proven.

Theorem 1.1 (Nies, Shore, Slaman [NSS98]). *Any relation on the c.e. degrees invariant under the double jump is definable in the c.e. degrees iff it is definable in first order arithmetic.*

The proof of Theorem 1.1 involves interpreting the standard model of arithmetic in the structure of the c.e. degrees without parameters, and a definable map from degrees to indices (in the model) which preserves the double jump. The beauty of this result is that it gives at one time a definition of a large class of relations on the c.e. degrees.

Theorem 1.1 has two shortcomings. One is the reliance on the invariance of the relation under the double jump. It follows that no set of c.e. degrees that contains some, but not all, low$_2$ (or non-low$_2$) degrees, can be defined using the theorem; these are the kinds of sets we investigate here.

Another issue is that the definitions provided by the theorem are not *natural* definitions of objects in computability theory (as outlined in Shore [Sho00].) Here we are thinking of results such as the following. (We refer the reader to Soare [Soa87] for unexplained definitions in the sequel, since they are used to provide background for the results of the current paper.)

Theorem 1.2 (Ambos-Spies, Jockusch, Shore, and Soare [ASJSS84]). *A c.e. degree* $\mathbf{a}$ *is promptly simple iff it is not cappable.*

The authors' research was supported by the Marsden Fund of New Zealand, via postdoctoral fellowships.

Theorem 1.3 (Downey and Lempp [DL97]). *A c.e. degree* $\mathbf{a}$ *is contiguous iff it is locally distributive, meaning that*

$$\forall \mathbf{a}_1, \mathbf{a}_2, \mathbf{b}(\mathbf{a}_1 \cup \mathbf{a}_2 = \mathbf{a} \wedge \mathbf{b} \leqslant \mathbf{a} \rightarrow \\ \exists \mathbf{b}_1, \mathbf{b}_2(\mathbf{b}_1 \cup \mathbf{b}_2 = \mathbf{b} \wedge \mathbf{b}_1 \leqslant \mathbf{a}_1 \wedge \mathbf{b}_2 \leqslant \mathbf{a}_2)),$$

holds in the c.e. degrees.

Theorem 1.4 (Ambos-Spies and Fejer [ASF01]). *A c.e. degree* $\mathbf{a}$ *is contiguous iff it is not the top of the non-modular 5 element lattice in the c.e. degrees.*

Theorem 1.5 (Downey and Shore [DS95]). *A c.e. truth table degree is low*$_2$ *iff it has no minimal cover in the c.e. truth table degrees.*

At the present time, as articulated in Shore [Sho00], there are very few such natural definability results.

In this paper, and in the sequel [DG], we will give some new natural definability results for the c.e. degrees. Moreover, these definability results will be related to the central topic of lattice embeddings into the c.e. degrees as analysed by, for instance, Lempp and Lerman [LL97], Lempp, Lerman and Solomon [LLS], and Lerman [Ler85]. Additionally our new definability results will allow us to tie a number of natural constructions together in new degree classes in the same way as the *array noncomputable* degrees did in Downey, Jockusch and Stob [DJS90, DJS96]. Here the reader should recall that a degree $\mathbf{a}$ is called array noncomputable iff for all functions $f \leqslant_{wtt} \emptyset'$ there is a a function g computable from $\mathbf{a}$ such that

$$\exists^\infty x(g(x) > f(x)).^1$$

To do this we will introduce a new degree class meant to capture a notion of "multiple permitting" which is stronger than array noncomputability, but weaker than non-low$_2$-ness.

Definition 1.6. We say that a c.e. degree $\mathbf{a}$ is *totally* ω*-c.e.* iff for all functions $g \leqslant_T \mathbf{a}$, g is ω-c.e. That is, there is a computable approximation $g(x) = \lim_s g(x, s)$, and a computable function h, such that for all x,

$$|\{s : g(x, s) \neq g(x, s + 1)\}| < h(x).$$

We remark that the key difference between being totally ω-c.e. and being array computable is that for the latter, the function $h(x) = x$ can always be chosen (Downey, Jockusch, Stob [DJS96]).

A central notion for lattice embeddings into the c.e. degrees is the notion of a *weak critical triple.* The reader should recall from Downey [Dow90] and Weinstein [Wei88] that three incomparable elements $\mathbf{a}_0, \mathbf{a}_1$ and $\mathbf{b}$ in an upper semilattice form a weak critical triple if $\mathbf{a}_0 \cup \mathbf{b} = \mathbf{a}_1 \cup \mathbf{b}$ and there is no $\mathbf{c} \leqslant \mathbf{a}_0, \mathbf{a}_1$ with $\mathbf{a}_0 \leqslant \mathbf{b} \cup \mathbf{c}$. We say that incomparable $\mathbf{a}_0, \mathbf{a}_1$ and $\mathbf{b}$ in an upper semilattice form a *critical triple* if $\mathbf{a}_0 \cup \mathbf{b} = \mathbf{a}_1 \cup \mathbf{b}$ and every $\mathbf{c}$ below both $\mathbf{a}_0$ and $\mathbf{a}_1$ is also below $\mathbf{b}$.[2] These notions

[1] Of course, this was not the original definition of array noncomputability, but this version from [DJS96] captures the domination property of the notion in a way that shows the way that it weakens the notion of non-low$_2$-ness, in that $\mathbf{a}$ would be non-low$_2$ using the same definition, but replacing $\leqslant_{wtt}$ by $\leqslant_T$.

[2] This definition becomes more natural in a lattice, where we can write $\mathbf{a}_0 \cap \mathbf{a}_1 \leqslant \mathbf{b}$.

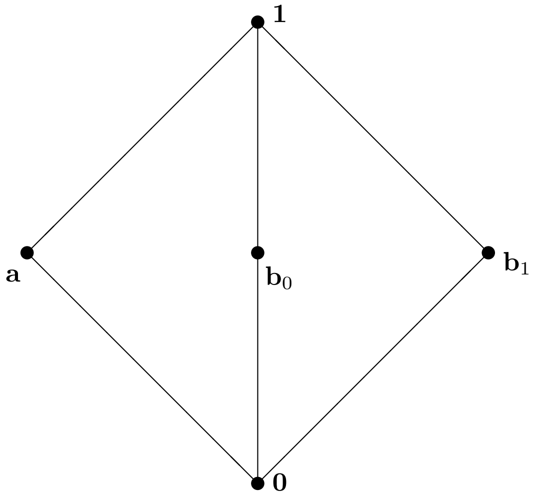

FIGURE 1. The lattice M_5

capture the need for "continuous tracing" which is used in an embedding of the lattice M_5 into the c.e. degrees (first embedded by Lachlan [Lac72]).[3]

Indeed the first nonembeddability result was by Lachlan and Soare [LS80] who demonstrated that an "infimum into an M_5" could not be embedded in the c.e. degrees by showing that the lattice S_8 below could not be embedded (as suggested by Lerman.)

The necessity of the "continuous tracing" process was further demonstrated by Downey [Dow90] and Weinstein [Wei88] who showed that there are initial segments of the c.e. degrees where no lattice with a (weak) critical triple can be embedded. It was also noted in Downey [Dow90] that the embedding of critical triples seemed to be tied up with multiple permitting in a way that was similar to non-low$_2$-ness. Indeed this intuition was verified by Downey and Shore [DS96] where it is shown that if **a** is non-low$_2$ then **a** bounds a copy of M_5.

The notion of non-low$_2$-ness seemed too strong to capture the class of degrees which bound M_5's but it was felt that something like that should suffice. On the other hand, Walk [Wal99] constructed a array noncomputable c.e. degree bounding no weak critical triple, and hence it was already known that array non-computability was not enough for such embeddings. We prove the following definitive results.

Theorem 1.7. *Suppose that* **a** *is totally* ω*-c.e. Then* **a** *bounds no weak critical triple.*

Theorem 1.8. *Suppose that* **a** *is not totally* ω*-c.e. Then* **a** *bounds a weak critical triple.*

Corollary 1.9. *The following are equivalent for a c.e. degree* **a***:*

(1) **a** *bounds a critical triple;*
(2) **a** *bounds a weak critical triple;*
(3) **a** *is not totally* ω*-c.e.*

[3] We recall that a lattice is not join semidistributive (also called principally indecomposable) iff it contains a copy of M_5 iff it contains a critical triple iff it contains a weak critical triple.

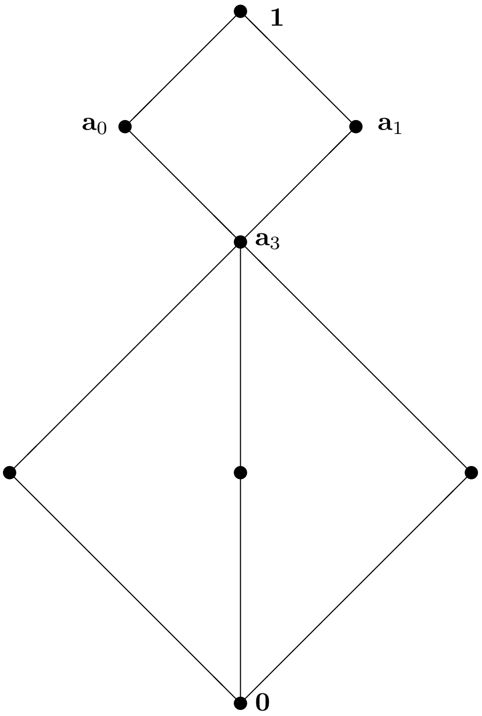

FIGURE 2. The lattice S_8

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

This result shows also that the array computable c.e. degrees form a proper subset of the totally ω-c.e. degrees:

Corollary 1.10. *There are c.e. degrees that are totally ω-c.e. and not array computable.*

Proof. Walk [Wal99] constructed an array noncomputable degree $\mathbf{a}$ below which there was no weak critical triple. Such a degree must be totally ω-c.e. □

Another corollary answers a question of André Nies. Recall that a set A is called *superlow* iff $A' \equiv_{tt} \emptyset'$.

Corollary 1.11. *The low degrees and the superlow degrees are not elementarily equivalent.*

Proof. As Schaeffer [Sch98] and Walk [Wal99] observe, all superlow degrees are array computable, and hence totally ω-c.e. Thus we cannot put a copy of M_5 below one. On the other hand there are indeed low copies of M_5. □

Finally, we observe that the class of totally ω-c.e. degrees captures another construction aside from the (weak) critical triple one. This construction ([DS86, Theorem 4.8]) too uses a variant of "continuous tracing". We can make two definitions that are analogous to the weak and strong varieties of critical triples.

Definition 1.12. Three c.e. sets A_0, A_1 and B form a *wtt triple* if $A_0 \equiv_T A_1$, $A_i \nleqslant_T B$, and for all $C \leqslant_{wtt} A_0, A_1$ we have $C \leqslant_{wtt} B$. The sets A_0, A_1, B form a *weak wtt triple* if $A_0 \equiv_T A_1$, $A_i \not\leqslant_T B$, and there is no $C \leqslant_{wtt} A_0, A_1$ such that $A_0 \leqslant_T B \oplus C$.

Theorem 1.13. *The following are equivalent for a c.e. degree* $\mathbf{a}$*:*

1. *There are* $A_0, A_1, B \leqslant_T \mathbf{a}$ *which form a wtt triple;*
2. *There are* $A_0, A_1, B \leqslant_T \mathbf{a}$ *which form a weak wtt triple;*
3. $\mathbf{a}$ *is not totally* ω*-c.e.*

In the sequel to the present paper, Downey and Greenberg [DG] introduce another (similar) class of degrees which capture a very wide class of constructions and turn out to be exactly the class of degrees which bound copies of M_5.

2. Constructions of critical triples

The goal of this section is to give a proof of Theorem 1.8. Recall how Downey and Shore's proof ([DS96]), that the M_5 can be embedded below any non-low$_2$ degree, is an elaboration on Lachlan's original construction of an embedding of M_5 into the c.e. degrees; working below a degree that permits sufficiently, they add a bottom set which codes these permissions. We aim to pursue a similar strategy; so first we need to give a "pure" construction of a critical triple.

2.1. A degenerate critical triple.

Unlike M_5, when insisting that a bottom of a critical triple $\mathbf{a}_0 \cap \mathbf{a}_1$ is $\mathbf{0}$, we get an object much simpler than the general notion. Nevertheless the construction needs the continuous tracing that is central to the class of constructions under consideration.

Theorem 2.1. *There are incomparable c.e. degrees* $\mathbf{a}_0, \mathbf{a}_1$ *and* $\mathbf{b}$ *such that* $\mathbf{a}_0$ *and* $\mathbf{a}_1$ *form a minimal pair, and such that for* $i < 2$ *we have* $\mathbf{a}_i \leqslant \mathbf{b} \cup \mathbf{a}_{1-i}$.

To prove the theorem, we enumerate sets A_0, A_1 and B that meet the following requirements:

N_Φ**:** If $\Phi(A_0) = \Phi(A_1) = Z$ are total and equal then Z is computable.
$P_{\Psi,i}$**:** $\Psi(B) \neq A_i$ $(i < 2)$.

Globally, we need to ensure that for $i < 2$, $A_i \leqslant_T B \oplus A_{1-i}$.

Remark 2.2. Meeting the requirements ensures that the sets are all incomparable. We have $A_i \not\leqslant_T B$, so neither A_i is computable; so they form a minimal pair (and in particular are incomparable). We cannot have $B \leqslant_T A_i$ or we'd have $A_{1-i} \leqslant_T B \oplus A_i \equiv_T A_i$, which we ruled out.

Discussion. Our starting point is the standard tree construction of a minimal pair (see [Soa87]); nodes (agents) that work for some N_Φ requirement ensure that "dangerous" numbers cannot enter both A_0 and A_1 at the same stage, by allowing only one number to be enumerated into A_0 or A_1 during an expansionary stage.

The global requirements are met using a tracing procedure. Thus, any number that is appointed and targeted to enter a set A_i, must have a *trace* which is targeted for either A_{1-i} or for B. Suppose that x is a follower for some node which works for a requirement $P_{\Psi,i}$. Until x is realized ($\Psi(B,x)\downarrow = 0$), we cannot appoint a trace for B, since it would be smaller than the anticipated Ψ-use. Hence a trace must be appointed for A_{1-i}. That trace, in turn, must get a trace for A_i, and so on; we get an *entourage* that consists of x and its descendants via tracehood. When x is realized, we want to enumerate x and its entourage into the target sets; but since the targets are alternating $A_i, A_{1-i}, A_i, \dots$, we need to peel the entourage off one at a time, so that the restriction imposed by minimality requirements is obeyed. Of course we need to keep appointing traces; if we keep appointing for A_i and A_{1-i}, we'll never get to peel off all of the entourage. But since x is realised, we can now appoint all traces for B.

Construction. We use a tree of strategies. Each level of the tree is devoted to meeting one requirement. Nodes ρ that work for some N_Φ have two successors, $\rho^\frown\infty$ and $\rho^\frown f$ (the former is stronger). Nodes that work for a P-requirement have only one successor.

At stage s, inductively construct the path of accessible nodes.

Suppose that ρ is an accessible node that works for N_Φ. Let the

$$\ell(\Phi)[s] = \max\{x : \forall y < x \ (\Phi(A_0, y)\downarrow = \Phi(A_1, y) \ [s])\}$$

be the *length of agreement* between $\Phi(A_0)$ and $\Phi(A_1)$ at s. If $\ell(\Phi)[s]$ is greater than the last expansionary stage t then s is declared to be expansionary, and we take the infinite outcome (i.e. $\rho^\frown\infty$ is accessible). Otherwise, we take the finite outcome ($\rho^\frown f$ is accessible).

Suppose now that η is an accessible node that works for $P_{\Psi,i}$. If η is *satisfied*, or if it has a follower x which is not realized, let the unique successor of η on the tree be accessible. Otherwise (we say η *requires attention*), we act as follows and then halt the stage.

- If η has no follower (this is the first time we visit η since it was last initialised), then we appoint a fresh follower x for η.
- If η has a realized follower, let x be η's follower, and suppose that x's entourage is $x = x_0, x_1, \dots, x_k$. If x_k is targeted for B, enumerate it into B, and enumerate x_{k-1} into the set for which it is targeted. Otherwise, just enumerate x_k into the set for which it is targeted (this is the first stage at which we see x realized). In either case, appoint a new, large, last trace to x's entourage (to be either x_{k-1} or x_k), and target it for B. If, however, we just enumerated x into A_i, then no new trace is defined; rather, η declares itself satisfied.

At the end of the stage, initialize all nodes that are weaker than the last accessible node (including its extensions.) [If a node η working for some P-requirement is initialised, it abandons its follower and all of its entourage, and declares itself unsatisfied.] Finally, for every entourage (for any node on the tree) whose last element y is targeted for some A_j, we add a new (large) trace for y at the end, and target it for A_{1-j}.

Verifications. As usual, the *true path* is the path of nodes that are leftmost with respect to being accessible infinitely often.

Lemma 2.3. *Suppose that η is a node on the true path that works for $P_{\Psi,i}$. If there is a stage after which η is not initialised, then η meets its requirement and requires attention only finitely often.*

Proof. Let r^* be the last stage at which η is initialised. Let s_0 be the least stage after r^* at which η is visited. At stage s_0 we appoint a follower x for η; x is never cancelled. If we never see x realised during a stage at which η is accessible, then η never requires attention again and the requirement is met.

Suppose that at some stage $s_1 > s_0$, η is accessible and sees x realised. The last element of x's entourage is enumerated into its target set, and the entourage receives a last trace x_k targeted for B. We first see that the computation $\Psi(B,x)[s_1]$ realizing x is correct. This is because nodes that are weaker than η are initialised at stage s_1, and later traces appointed are larger than the use; and nodes stronger than η don't enumerate numbers into B after stage r^*. Finally, all the traces that η appoints that are targeted for B are appointed no earlier than stage s_1 and so too are larger than the use.

At subsequent stages, whenever η is accessible, it is still realised, and the two current last elements of x's entourage are enumerated into their target sets. Thus at each such stage, the length of x's entourage shrinks by 1. Eventually x will be enumerated into A_i and η becomes satisfied, and never requires attention again. □

As a result, by induction we can see that every P-node on the true path is not initialised infinitely often, and meets its requirement.

Lemma 2.4. *Every N_Φ requirement is met.*

Proof. Let ρ be a node on the true path that works for N_Φ. Assume that $\Phi(A_0) = \Phi(A_1) = Z$ are total and equal. The familiar argument for showing that Z is computable follows since η is eventually never initialised; at every η-expansionary stage, at most one number is enumerated into either A_0 or A_1; and nodes that enumerate numbers into sets between expansionary stages are initialised during expansionary stages. □

Lemma 2.5. *For $i < 2$, $A_i \leqslant_T B \oplus A_{1-i}$.*

Proof. Consider any $x < \omega$. If at stage $x+1$, x is not appointed as an element of some entourage which is targeted for A_i, then $x \notin A_i$. Otherwise, at stage $x+1$, x is appointed a trace y_0, targeted for B or for A_{1-i}. If it is targeted for B, then $x \in A_i$ iff $y_0 \in B$. If it is targeted for A_{1-i}, and $y_0 \notin A_{1-i}$, then $x \notin A_i$. Otherwise, we can find a stage at which y_0 is enumerated into A_{1-i}; at that stage, x is appointed a trace y_1 targeted for B, and again $x \in A_i$ iff $y_1 \in B$. □

2.2. **The proof of Theorem 1.8.** Let D be a c.e. set that computes a function g which is not ω-c.e.; let Γ be a Turing functional such that $\Gamma(D) = g$. We enumerate sets B, A_0, A_1. The requirements to meet are:

N_Φ**:** If $\Phi(A_0) = \Phi(A_1) = Z$ are total and equal then $Z \leqslant_T B$.

$P_{\Psi,i}$**:** $\Psi(B) \neq A_i$ $(i < 2)$.

We also ensure that for $i < 2$, $A_i \leqslant_T B \oplus A_{1-i}$. Permitting will be used to ensure $A_i, B \leqslant_T D$.

Remark 2.6. The requirements ensure that the degrees of A_i and B form a critical triple. To see this we just need to see that B, A_0 and A_1 are all incomparable. We cannot have $A_i \leqslant_T A_{1-i}$; for then we'd have $A_i \leqslant_T A_i, A_{1-i}$ and so $A_i \leqslant_T B$, which we ruled out. But then (as before), we also cannot have $B \leqslant_T A_i$.

Discussion. We now describe how to add permitting to the previous construction. Once we decide how to grant permissions, the mechanism of implementing this decision is familiar: instead of enumerating a pair of numbers (x, y) (where x is targeted for some A_i and its trace y is targeted for B) into their target sets, we first place the pair in a *permitting bin* and await for permission. Only when permitted (and since permissions will not be granted cofinally, at the stage at which x is permitted, the node η that appointed x may not be accessible) do we enumerate the pair; and then appoint a new B-trace to the end of the remaining entourage. Since the pair may never get permission, the node η must in the meantime appoint more and more followers and hope to succeed on one of them.

The difficulty lies in the fact that to succeed on a follower x, η must receive *several* permissions for x. This is why simple permitting will not succeed: even though in general, η will receive permission infinitely many times, so infinitely many followers will receive permission at least once, it is possible that no follower will receive enough permissions to enumerate all of its traces. [So if we get confinally many permissions, as in high permitting, the construction works in a straightforward way. This shows that every high degree bounds a critical triple (which is of course not new).] Also, we cannot bound in advance how many permissions each follower needs; we need to wait until a follower is realised to know how long the eventual entourage is going to be (as we soon see, if we could bound this length in advance, then every array noncomputable degree would bound a critical triple).

We want to argue that if no follower receives enough permissions then we can approximate g in an ω-c.e. fashion. To do this, every follower x receives a *permitting number* n (say, the n^{th} follower appointed receives the number n.) We have a Δ^0_2-approximation $g[s]$ for g that is generated from an enumeration of D via Γ; thus changes in the approximation are tied to D-changes. We wait until x is realized; we then know the number of permissions we need. This is computable (given n). And so we can permit x whenever there is a change in our guess for $g(n)$. If not enough permissions are given then we know a bound on how many times this guess can change. Note that we use the fact that D is c.e., even though the permission are granted via $g[s]$-changes; to know when permissions will no longer be granted, D needs to know that a correct $\Gamma(n)$-computation is permanent.

There is one last complication. Whenever we act for a follower x, we need to cancel all weaker (larger) followers for the same node η; this is because the computation realizing a weaker follower may have use which is greater than a B-trace on x's entourage. This cancellation leaves gaps in our threatened approximation for g; we may receive all the permissions on permitting numbers that are assigned to cancelled followers. The solution is for the cancelling follower x to assume responsibility for approximating g on all of the permitting numbers of the followers that were cancelled because of x (sometimes you have to pay even for inflicting finite injury). To make this work we need to incorporate the number of permissions required by x into the bound on the number of changes for $g[s]$ that the weaker follower promises. Now, x receives the permissions that are granted to cancelled followers (so the price paid buys quite a lot).

Construction. For each follower x (appointed by some P-node η), we associate a *permission interval* $I(x)$. This is an interval of natural numbers; its left end is fixed from the time at which x is appointed, but its right end may grow with time (but only finitely often). Suppose that at some stage s, elements of x's entourage are placed in the permitting bin, and that at stage $t > s$, these elements are still in the bin. Then we say that x is *permitted* at stage t if for some $n \in I(x)[t]$ (we will in fact have $I(x)[t] = I(x)[s]$), we have $g(n)[s] \neq g(n)[t]$. Here $g[s]$ is a Δ^0_2 approximation for g that is obtained by a computable enumeration of D via applying Γ.

At the beginning of stage s, if there are any permitted followers, then let x be the strongest one. Let η be the node that appointed x. We enumerate those elements of x's entourage that lie in the permitting bin into their target sets. We initialise all nodes weaker than η.

- If the follower x has not been just enumerated, then we appoint a new B-trace at the end of x's remaining entourage. We cancel all η-followers y that are weaker (greater) than x. For each such y, redefine $I(x) := I(x) \cup I(y)$.
- If the follower x has just been enumerated, then all followers for η are cancelled, and η is declared to be satisfied.

Next, we construct the path of accessible nodes.

Suppose that ρ is an accessible node that works for N_Φ. We define the length of agreement and expansionary stages exactly as in the previous subsection; we act the same.

Suppose that η is an accessible node that works for $P_{\Psi,i}$. If η is satisfied, or if it has an unrealized follower, then η does nothing and its only child is accessible.

Otherwise:

- If all followers for η have some numbers waiting in the permitting bin (this includes the case that there are no followers appointed), then η appoints a new (large) follower x. Let n be the least number that is not in $I(y)$ for any other η-follower y (x will be the $(n+1)^{\text{st}}$ follower appointed for η since the last time at which η was initialised,) and define $I(x) = \{n\}$.
- Suppose that there is a follower x that is realized, but no elements of x's entourage lie in the permitting bin (necessarily x will be η's weakest follower.) If the last element of x's entourage is not targeted for B, appoint a new B-trace for the end of the entourage (this happens only at the first time at which we see x realized). Place the last two elements of x's entourage in the permitting bin.

At the end of the stage, initialise nodes weaker than those accessible. For every entourage whose last element is targeted for some A_i, appoint a new end trace, targeted for A_{1-i}.

Verifications. Define the true path as usual.

Lemma 2.7. *Suppose that η is a P-node on the true path, and suppose that after some stage, η is not initialised. Then η requires attention finitely often and meets its requirement.*

Proof. Let r^* be the last stage at which η is initialised.

It is easy to see (by induction on the stages) that at any stage, all but η's weakest follower are realized and have elements of their entourage waiting in the permitting bin. Also, if at some point η is declared satisfied then indeed the requirement is met; this is exactly as before.

Suppose that η requires attention infinitely often. Every follower that η appoints receives attention only finitely many times. The argument is as before: after being realized, a follower's entourage keeps shrinking (in this construction, it shrinks after *two* times it receives attention: once to put elements in the permitting bin, and then when they get enumerated). Indeed, if we let m_x be the length of a follower x's entourage when it receives its first B-trace, then x receives attention at most $2m_x + 1$ many times. So there are infinitely many followers appointed.

So for every $n < \omega$ there is a follower x_n which when appointed receives n as its permitting number.

We define an approximation for g as follows: at stage $s > r^*$, if some follower x for η puts numbers into the permitting bin, and $n \in I_\eta(x)[s]$, then guess that $g(n)$ is $g(n)[s]$.

Every follower that is appointed is either cancelled or eventually realized (and equipped with a B-trace). Otherwise, some x is appointed and never realized when η is accessible (and never cancelled); after the stage at which x was appointed, η never receives attention.

So to bound the number of changes we make on our guess for $g(n)$, go to the least stage s at which x_n is either cancelled or receives a B-trace. Then after s we do not make more than $\sum m_y$ guesses for $g(n)$, where y ranges over followers for η at stage s. This is effective.

The approximation is correct. There is a smallest follower x such that n ever enters the permitting interval $I_\eta(x)$; say after stage t_n we always have $n \in I_\eta(x)$. The follower x is never enumerated into A_i, because in that case η would be satisfied. Also, it is never cancelled. The node η is accessible unboundedly many times, and so the only possible final outcome for x is that at some last stage s_x, some elements of x's entourage are placed in the permitting bin and x is never permitted later. At stage s_x we guess that $g(n) = g(n)[s_x]$ and we never change our guess. The value must be correct because otherwise there is some $t > s_x$ such that $g(n)[s_x] \neq g(n)[t]$; but then x is permitted at t.

Thus we get an ω-c.e. approximation for g; this is a contradiction. □

Again, no node on the true path is initialised infinitely many times. To see this we need to note that if σ is a node on the true path, then only finitely many nodes to its left are ever accessible; each one puts finitely many numbers into the permitting bin, and so eventually, none of these nodes initialises σ.

Lemma 2.8. *$A_i, B \leqslant_T D$.*

Proof. Work effectively with oracle D. To find if some number y will ever enter a set A_i or B, go to stage y and see if it is part of some entourage of a follower x for some node η. If not, then y does not enter any set.

Let $t_0 = y$. For every $n \in I_\eta(x)[t_0]$ we can find a stage s_n at which the computation $\Gamma(D, n)$ is correct; n will not be a cause for permission for x after stage s_n (here we use the fact that D is c.e.). Of course, it is possible that at $t_1 = \max\{s_n : n \in I_\eta(x)[t_0]\}$ we have $I_\eta(x)[t_1] \neq I_\eta(x)[t_0]$. But then we repeat the process and let $t_2 = \max\{s_n : n \in I_\eta(x)[t_1]\}$. Since $I_\eta(x)$ only grows finitely many times (when x receives attention), we eventually get some stage t^* that has the property $t^* \geqslant \max\{s_n : n \in I_\eta(x)[t^*]\}$ (we assume that x is not cancelled before t^*.) We show that no numbers of x's entourage are enumerated into any sets after stage t^*.

Suppose that at the beginning of stage t^*, there is some part of x's entourage waiting in the permitting bin. By induction on $s \geqslant t^*$ we show that x does not receive attention at s. If $s \geqslant t^*$ is the least stage at which x receives attention, then we know that $I_\eta(x)[s] = I_\eta(x)[t^*]$. But at s, x must be permitted via some $n \in I_\eta(x)$, contradicting $s \geqslant t^* \geqslant s_n$.

Another possibility is that at t^*, x still has no B-trace. But then we know that x is the last follower appointed by η up to stage t^*. If at some stage $s > t^*$, x

will receive attention, then since there will be no weaker followers for η at s we will still have $I_\eta(x)[s+1] = I_\eta(x)[t^*]$. At s, elements of x's entourage are put in the permitting bin. Now the same inductive argument shows that x never receives attention after s, and so x does not enumerate numbers into sets after t^*.

Finally it is possible that at t^*, x has a B-trace, but the entire entourage is waiting at the node η. Let $r \leqslant t^*$ be the last stage up to t^* at which x received attention (at r, some elements of x's entourage were permitted.) Then η is not accessible between r and t^*, because at the next stage after r at which η is accessible some follower $y \leqslant x$ for η must receive attention. It follows that at t^*, x is the weakest follower of η. Again the inductive argument above shows that x never enumerates any numbers into any set after t^*. □

For $i < 2$, $A_i \leqslant_T B \oplus A_{1-i}$; the argument is exactly as in the previous subsection.

Lemma 2.9. *Every N-requirement is met.*

Proof. Suppose that the hypothesis of N_Φ holds. Let ρ be the node on the true path that works for N_Φ. We know that $\rho^\frown\infty$ is on the true path (there are infinitely many ρ-expansionary stages).

A ρ-expansionary stage s is called *ρ-good* if no number waiting in the permitting bin at stage s (more precisely, at the second part of the stage at which the accessible nodes are calculated, after numbers may have entered sets but before numbers are placed in the bin) will ever be enumerated into a set. Note that B can effectively recognise good stages, because all numbers in the bin have B-traces.

We show that there are infinitely many stages that are ρ-good. Let s^* be a stage after which ρ is never initialised. Let $s > s^*$ be any stage; let x be the smallest number active at s that will ever receive attention after stage s. Suppose that x last receives attention at stage $t > s$; let $r > t$ be the next ρ-expansionary stage after t. We claim that r is good. For suppose that y is a number that lies in the permitting bin at stage r. If $y < x$ then y must have been appointed before x was; but this means that y is active at stage s and so by minimality of x, y does not receive attention after s. If $y > x$ then y must have been appointed after stage t; it was not appointed by a node stronger than ρ because $t > s^*$. Since $\rho^\frown\infty$ was not accessible between t and r, y was appointed by some node that lies to the right of $\rho^\frown\infty$ and so y is cancelled at stage r.

Now suppose that $s > s^*$ is a good stage, and let $m < \ell(\Phi)[s]$. We show that there is some good stage $t > s$ and some $i < 2$ such that the computation $\Phi(A_i, m)[s]$ is preserved between s and t, so the Lachlan minimal pair algorithm works to show that $Z \leqslant_T B$.

At stage s, at most one number targeted for some A_j is placed in the permitting bin. If at stage s no number is placed in the permitting bin, or if a number that will not be later enumerated into a set is placed in the bin, then the next ρ-expansionary stage t must be good, because at t, all numbers put in the bin between s and t are cancelled. Also, suppose that a number y enters either A_j between t and s. y cannot be in the bin at s because s is good. Also, y cannot originate from above $\rho^\frown\infty$ because it wasn't put in the bin at s and these nodes are not accessible until t. So y originated from a node that lies to the right of $\rho^\frown\infty$, and as these were initialised at s, y must have been appointed after s and so is greater than s, and so is greater than the use $\phi(A_i, m)[s]$.

For the remaining case, suppose that a number x, targeted for A_j, enters the bin at stage s, and enters A_j at stage $r > s$. Let $t \geqslant r$ be the next ρ-expansionary stage and let $i = 1 - j$. We claim that t and i are as required.

First, we show that no number $y < s$ enters A_i between stages s and t. Suppose it does. We must have $y < x$, because numbers greater than x are cancelled at s and new numbers greater than x are appointed greater than s. Because s is good, y is not in the bin at stage s; so y must originate from some node η extending $\rho^\frown\infty$, and must still be at η at stage r; but then y cannot be put in the bin before stage t. (Note that this argument holds if y belongs to an entourage stronger than x but also if y belongs to the same entourage as x.)

Next, we show that t is good. In fact, every number y that is waiting in the bin at stage t (when $\eta^\frown\infty$ is visited) was already in the bin at stage s. For let y be a number that is not in the bin at stage s but is at stage t. Since η is not initialised after stage s, y cannot be appointed by a node which is stronger than η. If it is appointed by some node that lies to the right of $\eta^\frown\infty$ then y is cancelled at t. Thus y is appointed by some node extending $\eta^\frown\infty$; so y cannot be placed in the bin between stage r and t, so must enter between s and r. But then, if y is weaker than x then it is cancelled at stage r; and if y is stronger than x then it would cancel x when it is placed in the bin. Also, y cannot be in the same entourage as x's, because no numbers from this entourage are placed in the bin between s and r. □

3. Degrees that do not bound critical triples

In this section we prove Theorem 1.7. We give a detailed proof of the following slightly weaker theorem:

Theorem 3.1. *Suppose that* $\mathbf{d}$ *is array computable. Then* $\mathbf{d}$ *bounds no weak critical triple.*

The proof of the general theorem would require only few modifications, which we describe later.

3.1. Discussion.

Let $D \in \mathbf{d}$ be c.e. Suppose that we are given $B, A_0, A_1 \leqslant_T D$ that are incomparable, and such that $A_i \leqslant_T B \oplus A_{1-i}$ $(i < 2)$. To show that they do not form a weak critical triple we need to construct a set $E \leqslant_T A_0, A_1$ such that $A_0 \leqslant_T B \oplus E$. The strategy is similar to that of [CDS98]. Recall that a fundamental notion is that of a *layer*, which we get by repeatedly applying the use function of the given reductions of A_0 and A_1 to the top of the triple. To preserve the correctness of a computation reducing $A_0(x)$ to $B\ oE$, we protect a certain amount of layers above x. The main argument is that to get a change in $A_0(x)$, there must be a sequence of changes in A_0 and A_1 starting at the last layer set up and working its way backwards (we say that a *layer is peeled* when such a change occurs.) The point is that if there is a change in either A_0 or A_1 below the last existing layer, then there is necessarily another change in the next layer, either in the other set (A_1 or A_0), or in B; in the first case, we can enumerate a marker into E and invalidate the reduction, and in the second case, the reduction is invalidated automatically. The argument then works because each node can calculate how much injury it needs to sustain from higher priority agents, and set up more layers; beyond the necessary injury, further changes are prevented by restraining the top set D that computes everything.

Now, of course, we have no control over D. The overall plan of this construction is to globally build a functional Γ such that $\Gamma(D)$ is total, and correlate peeling of layers to changes in $\Gamma(D)$'s values. Infinitely many agents, one for every attempt at computably approximating $\Gamma(D)$ in an id-c.e. fashion, build their version of E and the relevant reductions; an agent that looks at the correct approximation will give us the required set E.

In this construction, we do not have control over the bounding set D. The plan of this construction is to build a functional Γ such that $\Gamma(D)$ is total, and correlate peeling of layers to changes in $\Gamma(D)$'s values. Because $\mathbf{d}$ is array computable, $\Gamma(D)$ will be id-c.e., so we know that some approximation will give us a tight bound on the number of changes to a particular value of $\Gamma(D)$, and so on the number of layers that can be peeled. Such a nice approximation is not given (the recursion theorem cannot be applied, since there is no uniform way to go from an index for Γ to an index for the approximation of $\Gamma(D)$.) Thus, we have infinitely many agents, each working with its own guess for the approximation for $\Gamma(D)$; each will build its own version of E and the relevant reductions, and an agent with a correct approximation will give the required set E.

Layers. Fix functionals Λ and Ψ_i $(i < 2)$ such that $\Lambda(D) = B \oplus A_0 \oplus A_1$, and $\Psi_i(B \oplus A_{1-i}) = A_i$. Let $x < \omega$ and $i < 2$. We let $(x,i)' = \psi_i(B \oplus A_{1-i}, x)$. An iteration of this layer "derivative" is as expected: We let $(x,i)^{(0)} = x$, and $(x,i)^{(n+1)} = \big((x,i)^{(n)}, (n+i) \mod 2\big)'$. Modification of these notions by a finite stage is done, as usual, by appealing to functionals and sets as they appear at that stage. Thus $(x,i)'[s] = \psi_i(B \oplus A_{1-i}, x)[s]$, with the main idea being that a change in $A_i(x)$ after stage s necessitates a change in either B or A_{1-i} below $(x,i)'\,[s]$. And if also $(x,i)' < \operatorname{dom}\Lambda(D)\ [s]$ then such a change also necessitates a change in D below $\lambda((x,i)')\,[s]$.

Strictly speaking, $(x,i)'[s]$ is defined only for $x < \operatorname{dom}\Psi_i(B \oplus A_{1-i})[s]$. But as we know that $\Psi_i(B \oplus A_{1-i})$ is total, we can always speed up the enumeration of the given sets to obtain the next value of $(x,i)'$, or for that matter, of $\lambda((x,i)^{(n)})$ for any particular x, i, n we wish. This is assumed to happen whenever we appeal to any of these values. Such a speed-up may change the situation and prompt some other action. This does not result in stages that take infinitely long to perform, because at each stage, there are finitely many computations that are being maintained, and a speed-up is used to redefine them only. This may injure some other computations, but as long as the situation does not stabilize, no new definitions are made, so the same computation will not be injured twice at the same stage.

Guesses. For every $e < \omega$, agent e enumerates a set E_e and builds a functional Ξ_e, with the intention that $\Xi_e(B \oplus E_e) = A_0$. For $i < 2$, it builds a functional $\Theta_{e,i}$ with the intention that $\Theta_{e,i}(A_i) = E_e$. The agent guesses that Δ_e is an id-c.e. approximation for $\Gamma(D)$, where $\langle \Delta_e \rangle_{e<\omega}$ is an effective list of partial computable functions from ω^2 to 2.

Some common conventions can be enforced effectively: for example, we declare that $\Delta_e(n,0) = 0$ for every n; and that at every stage, $\operatorname{dom}\Delta_e$ is downward closed in both coordinates. Further, we can assume that each Δ_e is total by 'stretching' and delaying guesses until new guesses are discovered: we let $\Delta_e(n,s)$ be the last value among $\Delta_e(n,0), \Delta_e(n,1), \dots, \Delta_e(n,t)$ where $t < s$ is the greatest such that $\Delta_e(n,t)\downarrow\ [s]$.

But Δ_e may fail to be a correct approximation for deeper reasons:

- For some n, more than n many changes in the guess $\Delta_e(n, s)$ are observed (this of course includes the case that the limit of $\Delta_e(n, s)$ does not exist). Such failure will be manifested at some finite stage and can be effectively detected. When this happens, the agent is abandoned and does not take part in the construction any more.
- For some n, the value $\Gamma(D, n)$ is fixed but all registered guesses have the opposite value. This will never be discovered at some finite time. An agent waiting for the correct guess will not be abandoned, but will not receive attention after some stage.

There is little interaction between agents, since they each enumerate their own sets and functionals, except for Γ, which is global. However, every particular value of $\Gamma(D)$ will be defined by a unique agent so again there is no interaction there. For each e and $x < \omega$ we assign a *layer number* $n_e(x)$ such that:

- The map $(e, x) \mapsto n_e(x)$ is a bijection between ω^2 and ω;
- For any e, if $x < y$ then $n_e(x) < n_e(y)$.

The value $\Gamma(D, n_e(x))$ will be defined by agent e in relation with its attempts to define and maintain $\Xi_e(B \oplus E_e, x)$. To ensure that $\Gamma(D)$ is total, whenever an agent e is abandoned, for every x, we define $\Gamma(D, n_e(x)) = 0$ with use 0.

Even between two different inputs $x < y$ of Ξ_e there is not much interaction. The requirement $n_e(x) < n_e(y)$ will ensure that at any stage, the use of $\Xi_e(x)$ is not greater than the use of $\Xi_e(y)$, and so larger inputs y cannot undefine a $\Xi_e(x)$ computation.

3.2. **Construction.** At stage s, every unabandoned agent $e < s$ defines and maintains $\Xi_e(x)$ (for $x < s$) as follows. The least value of $x < s$ for which action is needed is always treated first.

If the computation $\Xi(B \oplus E, x)$ is undefined, agent e will execute the following algorithm.

How to define a new $\Xi(B \oplus E, x)$ computation:

Let $u = (x, 0)^{(n(x)+1)}$. Let v be a large number (so v was never used before by anyone; in particular $v \notin E$).

Define:

(1) $\Xi(B \oplus E, x) = A_0(x)$ with B-use u and E-use $v + 1$.
(2) $\Gamma(D, n(x)) = 0$ with use $\lambda(u)$.
(3) $\Theta_i(A_i, v) = 0$ with use u.

Further (to make $\Theta_i(A_i)$ total), for every $z < v$ that is not currently picked as some E-use for any $y < x$, we define (if not already defined) $\Theta_i(A_i, z) = E(z)$ with use 0.

We let $u(x) = u, v(x) = v$ and call u, v the *parameters* of the $\Xi(x)$ computation.

The agent next monitors the defined computations. Suppose that at some stage t_0, a $\Xi(x)$ computation is defined with parameters u, v. Let $n = n(x)$.

Suppose that at $s > t_0$ the computation still holds. The following will be evident from the construction.

- Let $i < 2$. If $\Theta_i(A_i, v)\downarrow$ $[s]$ then it does so with use u.

- If $\Gamma(D,n)[s]$ is defined then its use is $\lambda(u)[s]$. So a change in A_i or B below u (which makes either $\Xi(x)$ or $\Theta_i(v)$ undefined) means that $\Gamma(D,n)$ also becomes undefined.
- Suppose that $\Xi(y)$ is also defined at s (where $y > x$). Then it was last defined at a stage $t \geqslant t_0$, so $v(x) < v(y)$; and also $u(x) < u(y)$ (this relies on $n_e(x) < n_e(y)$ and some innocent properties of the given functionals Ψ_i and Λ, namely, uses go up in time and in input.)
- It follows that $\gamma(D,n(x)) < \gamma(D,n(y))$ $[s]$.

At each stage the agent acts according to which of D, A_0, A_1, B have changed below the use of the computation being monitored.

Trivial changes (case 1: only D changes). We correct $\Gamma(D,n)$ if it becomes undefined due to a trivial change. A trivial change (say from stage $s-1$ to stage s) is a change in D below $\lambda(u)[s-1]$ which is not accompanied by any change in either A_i or B below u. This change doesn't really affect input x, but $\Gamma(D,n)$ does become undefined; so in this case we redefine $\Gamma(D,n)[s]$ to have the same value as $\Gamma(D,n)[s-1]$ with new use $\lambda(u)[s]$.

A vanished computation (case 2: B changes). It is possible that at some stage s, $B \restriction u$ changes, making $\Xi(x)$ undefined. As mentioned, it follows that $\Gamma(D,n(x))$ is also undefined. A new $\Xi(x)$ computation is now defined according to the algorithm above. The old parameter v is abandoned and so no further corrections need to be made to $\Theta_i(A_i,v)$ (which may or may not be currently defined).

A voluntary vanishing (case 3: both A_i change). Suppose that *both* $\Theta(A_0,v)$ and $\Theta(A_1,v)$ are undefined at stage s. We then cancel the $\Xi(x)$ computation ourselves by enumerating v into E (both $\Theta_i(A_i,v)$ can be redefined correctly). We redefine $\Xi(x)$ as above.

- We need to ensure that in this situation too, $\Gamma(D,n)$ is undefined. But the most recent A_i change guarantees it.
- After such action, for all $y > x$, $\Xi(y)$ vanishes; and $\Gamma(D,n(y))$ is also undefined, so agent e can redefine all of these computations.

A layer is peeled (case 4: one A_i changes). Suppose that up to some stage t_1, all computations $\Xi(x)$ and $\Theta_i(v)$ are defined, but at t_1, a single A_i changes below u. When this happens, the input x is put on alert. The A_i change frees us to define $\Gamma(D,n)[t_1] = 1-\Gamma(D,n)[t_1-1]$. Now the agent waits until some stage $s > t_1$ at which we have

$$\Delta_e(n,s) = \Gamma(D,n)[s] \quad (= \Gamma(D,n)[t_1+1]) \ .$$

When such a Δ guess is discovered (the price for unpeeling the A_i layer is paid), the agent redefines $\Theta_i(A_i,v) = 0$ with use u.

- While x is waiting, we maintain $\Gamma(D,n)$ by attending to trivial changes in D.
- Of course, if, while x is on alert, a change in B or in A_{1-i} below u is discovered, then we find ourselves in one of the previous scenarii and we define a whole new $\Xi(x)$ computation.
- While e is waiting for a $\Delta(x)$ guess, it keeps maintaining $\Xi(y)$ computations for $y > x$. That's ok since $\Xi(x)$ is not cancelled. This is pure altruism: the agent wouldn't mind waiting, but we need $\Gamma(D)$ to be total.

We have surveyed all the possibilities and so this concludes the construction.

3.3. Verifications.

Lemma 3.2. *For any agent e and input x, $\Xi_e(x)$ is redefined only finitely many times.*

Proof. This is done by induction on x. Assume that after some stage s_0, agent e does not redefine $\Xi_e(y)$ for any $y < x$. Let $n = n_e(x)$ and let $u = (x, 0)^{(n+1)}$. Let $s_1 > s_0$ be a stage at which $B \restriction u$ and both $A_i \restriction u$ are correct. Suppose that at some stage $s_2 > s_1$, agent e makes a new $\Xi_e(x)$-definition; we then have $u(x)[s_2] = u$. This computation cannot be invalidated by a B-change, cannot be cancelled due to action for some $y < x$, and x will not be prompted to destroy it due to A_i changes. □

Corollary 3.3. $\Gamma(D)$ *is total.*

Proof. Let $n < \omega$; find e, x such that $n = n_e(x)$. If e is ever abandoned then $\Gamma(D, n)$ is permanently defined. By the previous lemma, there is a stage s_1 at which a permanent $\Xi_e(x)$ computation is defined, with parameters u, v. Find a stage $s_2 > s_1$ at which $D \restriction \lambda(u)$ is correct. If after stage s_2, $\Gamma(D, n)$ becomes undefined, then it gets redefined with use $\lambda(u)$ and cannot be injured later. □

Fix some e such that Δ_e is an id-c.e. approximation for $\Gamma(D)$. Agent e is never abandoned. We drop e-indices.

Lemma 3.4. $\Theta_i(A_i) = E$.

Proof. Let $v < \omega$. There are two cases. If at some stage s, v is not an E-parameter for any x, and a new $\Xi(x)$ computation is defined with larger $v(x)$, then at stage s, $\Theta_i(A_i, v)$ is defined with use 0 (so the computation is permanent), and value $E(v)[s]$. This value must be correct because v will not be picked later as any $v(y)$.

If this is not the case, then necessarily at some stage s, v is picked as some $v(x)$. We claim that x will define a permanent $\Theta_i(A_i, v)$ computation. For either the $\Xi(x)$ computation is later destroyed, in which case, if yet later $\Theta_i(A_i, v)\uparrow$ then we find ourselves back in case one. Or, $\Xi(x)$ is never cancelled; in which case some layers may be peeled (but no more than $n(x)$ many), but a last $\Theta_i(A_i, v)$ computation must be defined. The point is that agent e approximates $\Gamma(D)$ correctly and so it cannot be stuck for ever, waiting for a guess to materialize, while $\Theta_i(A_i, v)$ remains undefined.

The value is $0 = E(v)[s]$. If ever v enters E it is because of action for x (to destroy $\Xi(x)$) and as described during the construction, this is only done if both $\Theta_i(A_i, v)$ are undefined at the time; this again catapults us back to the first case. □

This is the heart of the argument:

Lemma 3.5. *Suppose that a $\Xi(x)$ computation is defined at some stage t_0 with parameters u, v, and let $t_1 < t_2 < \ldots t_k$ be the subsequent stages at which a $\Theta_i(A_i, v)$ computation is redefined (but at which the t_0 computation still holds). Let $l \leqslant k$; let $m = n(x) + 1 - l$, and let $i = m \mod 2$. Let $z = (x, 0)^{(m)}[t_0]$. Then*

$$A_i[t_l] \restriction z = A_i[t_0] \restriction z.$$

Proof. By induction on l. For $l = 0$ there is nothing to prove. Assume it holds for $l-1$; so for $w = (x,0)^{(m+1)}[t_0]$ and $j = 1 - i$ we have $A_j[t_{l-1}] \upharpoonright w = A_j[t_0] \upharpoonright w$ and so $w = (x,0)^{(m+1)}[t_{l-1}]$ (since there was no B-change after t_0). At some stage $s_l \in (t_{l-1}, t_l)$, we get a change in A_0 or A_1 below u. However, the change cannot be in $A_i \upharpoonright z$. This is because at stage t_{l-1}, $A_i \upharpoonright z \subseteq \Psi_i(B \oplus A_j)$ as the use is w. Any change in A_i below z necessitates a change in A_j below $w < u$ and so the destruction of $\Xi(x)$.

It follows that the change is in A_i above z or in A_j, (necessarily above $(x,0)^{(m-1)}$). In the first case, no further change in A_j is possible before t_l, and so there is no change in A_i below z either. In the second case, no further change in A_i is possible before t_l. So the induction is carried to t_l. □

Lemma 3.6. *$A_0 \leqslant_T B \oplus E$.*

Proof. For every x, a final $\Xi(x)$ computation is defined and is never altered, so $\Xi(B \oplus E)$ is total. Suppose that the final $\Xi(B \oplus E, x)$ computation is defined at stage s_0, with layer number n. We show that $A_0(x)[s_0] = A_0(x)$. Let $t_0 < t_1 < \dots$ be the stages at which case five occurs for x (after s_0), and let $z_k = (x,0)^{(k)}[s_0]$. By induction we show that if $k \leqslant n - l$ and $i = k \mod 2$ then $A_i \upharpoonright z_k$ did not change between s_0 and t_l. The conclusion then follows from the fact that case four can occur at most n times (or agent e would be abandoned). □

3.4. **The proof of theorem 1.7.** There is only one alteration to the previous proof that we need to add on to get Theorem 1.7. Under the assumption that D is totally ω-c.e., we know that $\Gamma(D)$ has an ω-c.e. approximation. An individual agent e will guess that Δ_e is such an approximation, with a bound given by the computable function φ_e. So all we need to do is instead of setting up n many layers for an input x with $n_e(x) = n$, to set up $\varphi_e(n)$ many such layers.

The problem, of course, is that φ_e may be partial. Of course the correct e gives us a total function; but an agent with an incorrect guess cannot wait until $\varphi_e(n)$ converges, since this may make $\Gamma(D)$ partial. The solution is to use simple permitting. Upon the first time of setting up x, we define $\Gamma(D,n)$ with use n; we do not define a Ξ_e (or $\Theta_{e,i}$) computation. We keep maintaining $\Gamma(D,n)$ with use n until we get $\varphi_e(n)$ to converge. After that, we wait for a D-change below n. Once we get it, n is *permitted*, and we can set up a $\Xi_e(x)$ computation with sufficiently many layers. If φ_e is total, then unless it is abandoned, agent e will get infinitely many permissions, for otherwise we can compute D in the usual way.

A minute change we should make is to let $\Xi_e(x)$ compute $A_o \upharpoonright x$ rather than $A_0(x)$; this is because only infinitely many x's (rather than all of them) are permitted. This makes no difference to the argument because setting up enough layers prevents changes in A_0 *up to* x, rather than only in $A_0(x)$.

To keep the use of $\Xi_e(x)$ monotone, we declare that if, while waiting for an input x to be permitted, a larger input y is permitted, then we can abandon x and never define a $\Xi_e(x)$ computation. This is of course harmless. We can also define $\Gamma(D, n_e(x))$ with use 0.

4. The wtt structure of a Turing degree

Most of the ideas used for the proofs of either direction of Theorem 1.13 were already utilised in the previous sections, and so we give fewer details. In section 4.1 we show that part 3 of Theorem 1.13 implies parts 1 and 2 by showing that any degree which is not totally ω-c.e. bounds a wtt triple. Section 4.2 holds the remainder of the equivalence, that a degree which is totally ω-c.e. does not bound a weak wtt triple.

4.1. **Existence.**

Let D be c.e. and suppose that $g = \Gamma(D)$ is not ω-c.e. We enumerate sets A_0, A_1, B; the requirements to meet are:

- N_Φ**:** If $\Phi(A_0) = \Phi(A_1) = Z$ are total and equal then $Z \leqslant_{wtt} B$. Here Φ ranges over weak truth table functionals.
- $P_{\Psi,i}$**:** $\Psi(B) \neq A_i$ $(i < 2)$. Here Ψ ranges over Turing functionals.

We also ensure that $A_0 \equiv_T A_1$, and by permitting, that $A_0, A_1, B \leqslant_T D$.

Remark 4.1. The main difference between this construction and that of a critical triple is that we cannot appoint traces for B, as we want $A_i \leqslant_T A_{1-i}$, not $A_i \leqslant_T B \oplus A_{1-i}$. Thus the entourages will keep growing even after a follower is realised. However, since we do not have bottom $\mathbf{0}$, this construction does not require the full pinball machinery of [DS86] (and the strength of ω^ω). Suppose that at some stage s a follower x is realised; let $x = x_0, x_1, \dots, x_m$ be x's entourage at that time. Then at the k^{th} time after s at which x receives attention, we try to enumerate x_{m-k} and all of its traces into their sets in one go. Stronger N_Φ-nodes need to be alerted of this; thus before being enumerated, x_{m-k} will also be assigned a marker targeted for B. At stage s we know how many markers we need, and so we can bound their size; this yields a weak truth table reduction of $\Phi(A_0) = \Phi(A_1)$ to B.

Construction. The construction takes place on a tree of strategies. A node η which works for $P_{\Psi,i}$ will appoint *followers* x and wait for them to be *realised*, i.e. for $\Psi(B, x)\downarrow = 0$. Each follower (targeted for A_i) will have a *trace* targeted for A_{1-i}; the trace will have a trace of its own, targeted for A_i, and so on. On top of the traces, after a follower x is realised, an element of x's entourage may receive a B*-marker.* This aids in meeting the infimum requirements.

Each follower is assigned a *permitting interval* $I(x)$. Suppose that at some stage s, elements of a follower x's entourage are placed in the permitting bin, and that at stage $t > s$ these elements are still in the bin. Then x is *permitted* at stage t if for some $n \in I(x)[t]$ we have $g(n)[s] \neq g(n)[t]$.

At the beginning of a stage s (in particular, before we determine the path of accessible nodes), if there is some permitted follower x, then we let x be the strongest one. Let η be the node that appointed x. We enumerate those elements of x's entourage that lie in the permitting bin into their target sets. We initialise all nodes weaker than η (they may still be accessible during the stage).

- If the follower x has not been just enumerated, then we appoint a new B-marker for the last element of x's remaining entourage. The B-marker is the least unused number in $\omega^{[x]}$ greater than $s^*(x)$, where $s^*(x)$ is the stage at which x was first appointed a B-trace. We cancel all η-followers y that are weaker (greater) than x. For each such y, redefine $I(x) := I(x) \cup I(y)$.
- If the follower x has just been enumerated, then all followers for η are cancelled, and η is declared to be satisfied.

Next, we construct the path of accessible nodes. Suppose that a node ρ working for N_Φ is accessible. Let $\ell(\Phi)$ be the length of agreement between $\Phi(A_0)$ and $\Phi(A_1)$.

If $\ell(\Phi)$ is greater than the previous expansionary stage then s is expansionary and $\rho^\frown\infty$ is accessible. Otherwise, $\rho^\frown f$ is accessible.

Suppose that η is an accessible node that is working for $P_{\Psi,i}$. If η is satisfied, or if it has an unrealized follower, then η does nothing and its only child is accessible. Otherwise, η *requires attention*:

- If all followers for η have some numbers waiting in the permitting bin (this includes the case that there are no followers appointed), then η appoints a new (large) follower x. Let n be the least number that is not in $I(y)$ for any other η-follower y (x will be the $(n+1)^{\text{st}}$ follower appointed for η since the last time at which η was initialised,) and define $I(x) = \{n\}$.
- Suppose that there is a follower x that is realized, but no elements of x's entourage lie in the permitting bin (necessarily x will be η's weakest follower.)
  - If this is the first time at which η is accessible and x is realised, then no element of x's entourage has a B-marker. Appoint a new B-marker for the last element of x's entourage. The value of this marker is the least element of $\omega^{[x]}$ greater than $s^*(x) = s$.

  Let y be the element of x's entourage that has a B-marker. Drop y, its marker, and all subsequent traces (if they exist) into the permitting bin.

At the end of the stage, initialise nodes weaker than those accessible. For every uncancelled follower, appoint a new (large) trace for the last element of its entourage.

Verifications. The nodes on the true path are eventually never initialised, and eventually do not receive attention; this is exactly as argued for Lemma 2.7. It follows (noticing that B-markers are appointed larger than $s^*(x)$, hence larger than the use of the computation realising x,) that every P-requirement is met. Also, $A_0 \equiv_T A_1$; every number targeted for A_i receives at most two traces, each targeted for A_{1-i}. The argument for Lemma 2.8 shows that for every follower x appointed, D can compute a stage after which no number associated with x is ever enumerated into any set. That $A_0, A_1 \leqslant_T D$ follows immediately, because numbers targeted for these sets are always chosen large. But numbers targeted for B are always chosen from $\omega^{[x]}$ for some follower x and so the same algorithm gives a decision procedure for B.

Thus it remains to check that the N-requirements are met.

Lemma 4.2. *Every N-requirement is met.*

Proof. This is an elaboration on the proof of Lemma 2.9. Let Φ be a wtt-functional (with use function ϕ). Let ρ be the node on the true path that works for N_Φ. Assuming that $\Phi(A_0) = \Phi(A_1) = Z$, we know that there are infinitely many ρ-expansionary stages. Let r^* be a stage after which ρ is never initialised.

As before, we say that an expansionary stage s is *good* if (when $\rho^\frown\infty$ is visited at that stage) no numbers residing in the permitting bin will be later enumerated into sets. For $m < \omega$, we say that a stage s is *good for m* if it is good, if $\ell(\Phi) > m\,[s]$, and moreover, if whenever x is an uncancelled follower at s (again, when $\rho^\frown\infty$ is accessible and weaker nodes are initialised) with a B-marker $z < \phi(m)$, then $z \notin B$ (so no numbers of x's entourage are later enumerated into sets).

Similarly to the argument of Lemma 2.9, we can show that the value of $\Phi(A_i, m)$ at a stage that is good for m is correct (note that if $s < t$ are good stages and s

is good for m then t is good for m). The point is that if $\phi(m) < z$ (where z is a B-marker for some follower x), then whenever numbers from x's entourage are enumerated into sets, only the smallest number can injure either $\Phi(A_0, m)$ or $\Phi(A_1, m)$; this puts us exactly in the situation in Lemma 2.9.

Similarly we can argue that there are infinitely many good stages; and certainly almost all are good for a given m.

We just need to see why this gives us a *weak truth table* reduction of Z to B. For this, let s be the least expansionary stage greater than r^* such that $\ell(\Phi) > m\,[s]$. Let x be the strongest follower active at s such that at a later stage, numbers from x's entourage enter a set. Let $r > s$ be the last stage at which numbers from x's entourage enter sets, and let $t \geqslant r$ be the next ρ-expansionary stage. If there is no such x then let $t = s$. We claim that t is good for m, and moreover, that we can effectively bound the queries from B that are required to ensure that t is good for m by the state of affairs at stage s.

If x does not exist then $t = s$ is good for m (as no follower active at x will ever enumerate any number into a set) and of course all B-markers that need to be consulted are defined at stage s. So we may assume that x exists.

Let y be a follower, still active when $\rho^\frown\infty$ is accessible at t. First, suppose that y has a B-marker z and that $z < \phi(m)$. We know that the first B-marker z' for y was appointed before stage s (because $\phi(m) < s$); its value is the least element of $\omega^{[y]}$ which is greater than $s^*(y)$ (and $s^*(y) < s$). The length of y's entourage at stage $s*^(y)$ is again below s, and the number of new B-markers is bounded by this length; for each B-marker, we take the next element of $\omega^{[y]}$. It follows that z is bounded by, say, the $2s^{\text{th}}$ element of $\omega^{[y]}$. Further, we know that $z \notin B$, because otherwise x is stronger than y and so y would be cancelled at r. (If $x = y$ then no numbers of y's entourage are enumerated after stage r.)

Next, suppose that numbers from y's entourage lie in the permitting bin at stage t. Since y is issued by some η extending $\rho^\frown\infty$ (otherwise y is issued by some node stronger than ρ; there are finitely many such y, and they will never enter sets after r^*), we know that these numbers were placed in the bin before stage r, and so y is stronger than x. It follows that numbers of y's entourage do not enter sets after stage s (otherwise, minimality of x is contradicted); and that the numbers in the bin were placed in the bin before stage s and so have a small B-trace as before. The point is that even if y is stronger than x, then it cannot be the case that y is only realized much later than stage s (and thus has a large B-marker) but also has numbers in the bin at stage t. □

4.2. **Nonexistence.** We show that if a c.e. set D is totally ω-c.e. then there are no $A_0, A_1, B \leqslant_T D$ that form a weak wtt triple. This is very similar to the argument for Theorem 1.7; the layers now correspond to the reductions of A_0 to A_1 and vice-versa. We note that the reductions $\Theta_{e,i}$ we got in section 3 were actually wtt reductions already.

Suppose that we are given such A_0, A_1 and B; let Ψ_i (for $i < 2$) be such that $\Psi_i(A_{1-i}) = A_i$, and Λ be such that $\Lambda(D) = A_0 \oplus A_1 \oplus B$. We let $(x,i)' = \psi_i(A_{1-i}, x)$, and we iterate to get $(x,i)^{(n)}$. We construct a functional Γ and let agent e guess that Δ_e is an ω-c.e. approximation for $\Gamma(D)$, with computable bound φ_e. Agent e enumerates a set E_e and builds functionals Ξ_e and $\Theta_{e,i}$ so that $\Theta_{e,i}(A_i) = E_e$, and Ξ_e reduces A_0 to $B \oplus E_e$. For each e and x we assign a layer number $n_e(x)$ as before.

Construction. At stage s, every unabandoned agent $e < s$ defines and maintains $\Xi_e(x)$ (for $x < s$) as follows. Let $n = n_e(x)$. When the agent e first tends to x, it defines $\Gamma(D, n) = 0$ with use n. If D changes below n then a new definition, with same use, is made. This is maintained until $\varphi_e(n)$ converges. Then, the agent waits until x is *permitted* by D changing below n, freeing us to redefine $\Gamma(D, n)$ with large use. When this happens, we are ready to define $\Xi(x)$. [If, while waiting for x to be permitted, a larger number y is permitted, then all attempts to define $\Xi(x)$ are abandoned. All agent e does with regards to x is keep defining $\Gamma(D, n)$ (it may as well define it with use 0).]

Whenever a computation $\Xi_e(B \oplus E_e, x)$ is undefined, agent e will define it according to the following algorithm.

> Let $m = \varphi_e(n)$ and let $u = (x, 0)^{(m+1)}$. Let v be a large number.
> Define:
> (1) $\Xi_e(B \oplus E_e, x) = A_0 \restriction x$ with B-use u and E_e-use $v + 1$.
> (2) $\Gamma(D, n) = 0$ with use $\lambda(u)$.
> (3) $\Theta_{e,i}(A_i, v) = 0$ with use u.
> For every $z < v$ that is not currently picked as some E_e-use for any $y < x$, we define (if not already defined) $\Theta_{e,i}(A_i, z) = E_e(z)$ with use 0.

The agent now monitors the defined computations. Suppose that at some stage t_0, a $\Xi_e(x)$ computation is defined with parameters u, v. At stage $s > t_0$: if a *trivial change* occurs, i.e. a change in $D \restriction \lambda(u)$ without a change in A_0, A_1 or B below u, then $\Gamma(D, n)$ is redefined with use $\lambda(u)\,[s]$. If the *computaion vanishes*, i.e. there is a $B \restriction u$ change, then a new $\Xi_e(x)$ computation is defined (with new u, v parameters). If there is a *double change*: both $A_0 \restriction u$ and $A_1 \restriction u$ change, then we enumerate v into E_e and redefine a new $\Xi_e(x)$ computation. If a *layer is peeled*, i.e. there is a single $A_i \restriction u$ change, then $\Gamma(D, n)$ is redefined with new value, and we wait for a correct $\Delta_e(n)$ guess; only after the guess is obtained do we redefine $\Theta_{e,i}(A_i, v) = 0$ with use u.

Verifications. These are exactly as in section 3.

References

[ASF01] Klaus Ambos-Spies and Peter A. Fejer. Embeddings of N_5 and the contiguous degrees. *Ann. Pure Appl. Logic*, 112(2-3):151–188, 2001.

[ASJSS84] Klaus Ambos-Spies, Carl G. Jockusch, Jr., Richard A. Shore, and Robert I. Soare. An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees. *Trans. Amer. Math. Soc.*, 281(1):109–128, 1984.

[CDS98] Peter Cholak, Rod Downey, and Richard Shore. Intervals without critical triples. In *Logic Colloquium '95 (Haifa)*, volume 11 of *Lecture Notes Logic*, pages 17–43. Springer, Berlin, 1998.

[DG] R. Downey and N. Greenberg. Domination and definiability II : The ω^ω case. In preparation.

[DJS90] Rod Downey, Carl Jockusch, and Michael Stob. Array nonrecursive sets and multiple permitting arguments. In *Recursion theory week (Oberwolfach, 1989)*, volume 1432 of *Lecture Notes in Math.*, pages 141–173. Springer, Berlin, 1990.

[DJS96] Rod Downey, Carl G. Jockusch, and Michael Stob. Array nonrecursive degrees and genericity. In *Computability, enumerability, unsolvability*, volume 224 of *London Math. Soc. Lecture Note Ser.*, pages 93–104. Cambridge Univ. Press, Cambridge, 1996.

[DL97] Rodney G. Downey and Steffen Lempp. Contiguity and distributivity in the enumerable Turing degrees. *J. Symbolic Logic*, 62(4):1215–1240, 1997.

[Dow90] Rod Downey. Lattice nonembeddings and initial segments of the recursively enumerable degrees. *Ann. Pure Appl. Logic*, 49(2):97–119, 1990.

[DS86] R. G. Downey and M. Stob. Structural interactions of the recursively enumerable T- and w-degrees. *Ann. Pure Appl. Logic*, 31(2-3):205–236, 1986. Special issue: second Southeast Asian logic conference (Bangkok, 1984).

[DS95] Rod Downey and Richard A. Shore. Degree-theoretic definitions of the low_2 recursively enumerable sets. *J. Symbolic Logic*, 60(3):727–756, 1995.

[DS96] Rod Downey and Richard A. Shore. Lattice embeddings below a nonlow_2 recursively enumerable degree. *Israel J. Math.*, 94:221–246, 1996.

[Lac72] A. H. Lachlan. Embedding nondistributive lattices in the recursively enumerable degrees. In *Conference in Mathematical Logic—London '70 (Proc. Conf., Bedford Coll., London, 1970)*, pages 149–177. Lecture Notes in Math., Vol. 255. Springer, Berlin, 1972.

[Ler85] Manuel Lerman. The embedding problem for the recursively enumerable degrees. In *Recursion theory (Ithaca, N.Y., 1982)*, volume 42 of *Proc. Sympos. Pure Math.*, pages 13–20. Amer. Math. Soc., Providence, RI, 1985.

[LL97] Steffen Lempp and Manuel Lerman. A finite lattice without critical triple that cannot be embedded into the enumerable Turing degrees. *Ann. Pure Appl. Logic*, 87(2):167–185, 1997. Logic Colloquium '95 Haifa.

[LLS] S. Lempp, M. Lerman, and D.R. Solomon. Embedding finite lattices into the computably enumerable degrees - a status survey. To appear.

[LS80] A. H. Lachlan and R. I. Soare. Not every finite lattice is embeddable in the recursively enumerable degrees. *Adv. in Math.*, 37(1):74–82, 1980.

[NSS98] André Nies, Richard A. Shore, and Theodore A. Slaman. Interpretability and definability in the recursively enumerable degrees. *Proc. London Math. Soc. (3)*, 77(2):241–291, 1998.

[Sch98] Benjamin Schaeffer. Dynamic notions of genericity and array noncomputability. *Ann. Pure Appl. Logic*, 95(1-3):37–69, 1998.

[Sho00] Richard A. Shore. Natural definability in degree structures. In *Computability theory and its applications (Boulder, CO, 1999)*, volume 257 of *Contemp. Math.*, pages 255–271. Amer. Math. Soc., Providence, RI, 2000.

[Soa87] Robert I. Soare. *Recursively enumerable sets and degrees.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987. A study of computable functions and computably generated sets.

[Wal99] S. Walk. *Towards a definitioon of the array computable degrees.* PhD thesis, University of Notre Dame, 1999.

[Wei88] B. Weinstein. *On embeddings of the 1-3-1 lattice into the recursively enumerable degrees.* PhD thesis, University of California, Berkeley, 1988.

School of Mathematics, Statistics and Computer Science, Victoria University, P.O. Box 600, Wellington, New Zealand

E-mail address: `Rod.Downey@vuw.ac.nz`

School of Mathematics, Statistics and Computer Science, Victoria University, P.O. Box 600, Wellington, New Zealand

E-mail address: `greenberg@mcs.vuw.ac.nz`

Department of Mathematics, Dartmouth College, 6188 Bradley Hall Hanover, NH 03755-3551

E-mail address: `rweber@math.dartmouth.edu`