Notre Dame Journal of Formal Logic
Volume 38, Number 3, Summer 1997

Infima in the Recursively Enumerable Weak Truth Table Degrees

RICH BLAYLOCK, ROD DOWNEY, and STEFFEN LEMPP

Abstract We show that for every nontrivial r.e. wtt-degree **a**, there are r.e. wtt-degrees **b** and **c** incomparable to **a** such that the infimum of **a** and **b** exists but the infimum of **a** and **c** fails to exist. This shows in particular that there are no strongly noncappable r.e. wtt-degrees, in contrast to the situation in the r.e. Turing degrees.

1 Introduction and notation Weak truth table reducibility (w-reducibility) was first introduced by Friedberg and Rogers [8]. Intuitively, we say that a set A is w-reducible to a set B (written $A \leq_{\mathrm{w}} B$) if there is a Turing reduction from A to B and a recursive function f such that, for any x, the value $f(x)$ bounds the greatest number whose membership or nonmembership in B is used to determine $A(x)$.

Since w-reducibility is a stronger reducibility than Turing reducibility, each Turing degree can be partitioned into the w-degrees of its sets. Ladner and Sasso [11] showed that there exists a nonzero *contiguous* degree, that is, an r.e. Turing degree which contains a single r.e. w-degree. The existence of such contiguous degrees, as well as the *strongly contiguous* degrees introduced by Downey [6], has been used to establish numerous existence results (cf. [11], [13], [1]) in the r.e. Turing degrees by establishing the corresponding results in the r.e. w-degrees.

Our results here deal with infima in the r.e. w-degrees, thus continuing the investigations of Cohen [5], Ambos-Spies [3], and Fischer [7]. Cohen's result that every incomplete r.e. w-degree is w-branching and Fischer's result that some initial segments of the r.e. w-degrees are lattices indicate that infima are more common in the r.e. w-degrees than in the r.e. Turing degrees. We reinforce this notion.

After giving an elegant finite injury construction of a pair of r.e. Turing degrees with no infimum [9], Jockusch asked whether *every* nonrecursive incomplete r.e. Turing degree was half of a pair without infimum. Ambos-Spies [2] and Harrington (in an unpublished work) independently answered this question affirmatively by introducing *strongly noncappable* r.e. degrees. An r.e. Turing degree **a** is strongly

Received February 16, 1995

noncappable if no r.e. $\mathbf{b}|_{\mathrm{T}}\mathbf{a}$ has an infimum with $\mathbf{a}$. Ambos-Spies and Harrington showed that there is a strongly noncappable degree incomparable with any given nonrecursive incomplete r.e. degree.

In Theorem 2.1, we show that every nontrivial r.e. w-degree caps nontrivially, so that the only r.e. w-degrees analogous to strongly noncappable Turing degrees are trivial. This result gives yet another contrast between the r.e. w-degrees and the r.e. Turing degrees.

In Theorem 3.1, however, we show that no nontrivial r.e. w-degree caps with all r.e. w-degrees. Thus in the r.e. w-degrees, like the r.e. Turing degrees, every degree is half of a pair without infimum. We remark that if two r.e. w-degrees do have an infimum, then the infimum is itself r.e. (cf. [12], Exercise IX.3.5). Our notation is for the most part standard, as in Soare [12].

Let ω denote the set of natural numbers including zero. By number we mean an element of ω and by set we mean a subset of ω. We use 2^{ω} to denote the set of infinite sequences of 0's and 1's, and $2^{<\omega}$ for the set of finite sequences of 0's and 1's. For any set S and number j, we denote the set of elements of S which are strictly less than j by $S \upharpoonright j$. Fix a recursive bijection from ω^n to ω which is increasing in each argument, and let $\langle x_1, x_2, \ldots, x_n\rangle$ denote the image of the n-tuple $(x_1, x_2, \ldots, x_n)$ under this bijection. For a fixed j, $\omega^{[j]}$ denotes the set of all pairs of the form $\langle x, j\rangle$.

Let T_e denote the e^{th} oracle Turing machine in some effective listing of all oracle Turing machines, and let $\{e\}^A$ denote the partial function computed by T_e with oracle A. We write $\{e\}^A_s(x) = y$ if $x, y, e < s$, machine T_e computes $\{e\}^A(x) = y$ in less than s steps, and the largest number used in the computation is less than s. The use function, $\mathrm{use}(A; e, x, s)$, is the least number greater than all those used in the computation $\{e\}^A_s(x)$, if this computation is convergent, and 0 otherwise. The use function, $\mathrm{use}(A; e, x)$, is $\mathrm{use}(A; e, x, s)$ if $\{e\}^A_s$ is defined for some s, and is undefined otherwise. We use upper case Greek letters (Φ, Ψ, etc.) to denote w-reductions and their lower case counterparts (φ, ψ, etc.) to denote the corresponding use functionals.

2 Nontrivial capping in the r.e. w-degrees Here we show that every nontrivial r.e. w-degree caps nontrivially.

Theorem 2.1 *For any nonrecursive, w-incomplete r.e. w-degree* $\mathbf{c}$*, there is an r.e. w-degree* $\mathbf{a}|_w\mathbf{c}$ *such that the infimum* $\mathbf{a} \cap \mathbf{c}$ *exists.*

Proof: Let C be a set of the given w-degree $\mathbf{c}$ with recursive enumeration $\{C_s\}$. We will construct sets A and B such that $A|_{\mathrm{w}}C$ and the w-degree of B is the infimum of those of A and C. To make $A|_{\mathrm{w}}C$, we will satisfy for each w-reduction Φ the requirements

$$P_\Phi : A \neq \Phi^C$$

and

$$N_\Phi : C \neq \Phi^A.$$

To make the w-degree of B the infimum of those of A and C, we will satisfy for each w-reduction Φ the requirement

$$Q_\Phi : \Phi^A = \Phi^C = D \longrightarrow D \leq_{\mathrm{w}} B,$$

as well as the global requirements $B \leq_{\mathrm{w}} A$ and $B \leq_{\mathrm{w}} C$.

We will achieve both of these last reductions by permitting, that is, we will only enumerate a number x into B at a stage when numbers lesser or equal to x enter both A and C. In fact we will enforce this on the A side by enumerating x itself into A whenever we enumerate it into B.

To aid in satisfying the remaining requirements, we will use a priority tree, with each node on the tree devoted to a single requirement. The nodes devoted to requirements of the form Q_Φ, called *infimum nodes*, will be regarded as having two possible outcomes, according to whether the hypothesis of the requirement is true or not. Nodes devoted to requirements of the form P_Φ or N_Φ, called *coding nodes* and *preservation nodes*, respectively, will be regarded as having only a single outcome. Thus our priority tree T is a subtree of a binary tree.

To be more precise, choose any priority ordering of all the requirements, and define the tree by recursion on the length of its nodes. Assuming that T has been defined for nodes of length less than $|\alpha|$, let α be devoted to the least requirement not yet associated with any node $\tau \subset \alpha$. If this requirement is of the form Q_Φ, then α has two successors, $\alpha\hat{}0$ and $\alpha\hat{}1$. Otherwise, α has a single successor $\alpha\hat{}0$.

As usual, $[T]$ denotes the set $\{f \in 2^\omega : (\forall n)[f \upharpoonright n \in T]\}$ of all infinite paths through T. We order nodes on the priority tree as follows: for $\sigma, \tau \in T$,

(i) σ is to the *left* of τ ($\sigma <_L \tau$) if

$$(\exists \rho \in T)[\rho\hat{}0 \subseteq \sigma \ \ \& \ \ \rho\hat{}1 \subseteq \tau];$$

(ii) $\sigma \leq \tau$ if $\sigma <_L \tau$ or $\sigma \subseteq \tau$;
(iii) $\sigma < \tau$ if $\sigma \leq \tau$ and $\sigma \neq \tau$.

Fix an effective coding of the elements of T and denote the code number of a node $\tau \in T$ under this coding by $\#\tau$.

To meet a requirement like P_Φ, we will code K into A at stages when the length of agreement between A and Φ^C grows. The particular coding is not crucial, as long as we are able to code all of (or even cofinitely much of) K into A if this length continues to grow. Then, if A were equal to Φ^C, we would have $C \leq_{\mathrm{w}} A \leq_{\mathrm{w}} K$, a contradiction. Of course, this strategy for a node α devoted to P_Φ may need to respect "restraints" of various higher priority nodes devoted to N- or Q-type requirements. This will be accomplished by "resetting" α, which simply means starting the coding process over, using markers greater than the current stage number (which will be larger than any such restraints). Requirement P_Φ will eventually be satisfied since the node on the true path devoted to it will be reset only finitely often.

The strategy for a requirement of the form N_Φ is the Sacks preservation strategy. When the agreement between C and Φ^A grows to a new maximum m, we will try to prevent numbers less than the combined use $\max\{\varphi(x) : x < m\}$ from entering A. This will be done by resetting all lower priority coding and infimum nodes. If α is the node on the true path devoted to N_Φ, then higher priority coding nodes will be finitary, so the only danger to α comes from higher priority infimum nodes. We will arrange so that, after an α-expansionary stage, such an infimum node will enumerate a number into A only if a smaller number has already entered A. Thus the infimum node will not be allowed to cause the first "injury" to α.

Finally, the basic strategy for a requirement of the form Q_Φ is to try to maintain computations common to Φ^A and Φ^C. At any given stage of the construction, a node α devoted to Q_Φ will have a certain length of agreement between Φ^A and Φ^C. At stages when this length grows, we will arrange for α to have a "trace" z for every pair $\langle x, y\rangle$ such that x is below the length of agreement and y is below the use $\varphi(x)$. Node α cannot reset all lower priority nodes when its length of agreement grows, since this may happen infinitely often. Thus α must occasionally allow lower priority nodes to enumerate into A and destroy a computation $\Phi^A(x)$. If a number such as y later enters C before the next α-expansionary stage, then we will enumerate the trace z into B to "inform" B that both sides of the computation have been lost. As mentioned earlier, in this case we will simultaneously enumerate z into A to keep the reduction $B \leq_{\mathrm{w}} A$.

At each stage s of the construction, we will construct a string $f_s \in T$ of length s which will be our current approximation to the true path, that is, the path $f \in [T]$ for which $f \upharpoonright m = \liminf_s f_s \upharpoonright m$ for all m, in the sense that for $\beta = f \upharpoonright m$ we have

(a) $(\exists^{<\infty} s)[f_s <_L \beta]$ and

(b) $(\exists^{\infty} s)[\beta \subseteq f_s]$.

Given a node $\tau \in T$, a stage s is called a *τ-stage* if $\tau \subseteq f_s$. The set of all τ-stages is denoted by S^τ.

At various substages of the construction we will be enumerating elements into A. At each stage s, we will need to define a length of agreement $l(\tau, s)$ for each $\tau \subseteq f_s$ appropriate for the requirement to which τ is devoted. The value of $l(\tau, s)$ will depend on the elements enumerated into A up to the point at which we define $l(\tau, s)$. Thus for convenience, at any point during the construction, we let A^+ denote the set of elements enumerated into A up to that point.

In the following construction, to *reset* a coding node simply means to mark it as having been reset, while to reset an infimum node additionally means to cancel all of the uncancelled traces associated with it.

2.1 Construction of A and B

Stage $s = 0$.

Let $A_0 = B_0 = \varnothing$, and let f_0 be the empty string.

Stage $s + 1$.

2.1.1 Trace enumeration First consider each number $y \in C_{s+1} - C_s$. (Depending on the convention chosen, there might be at most one such number y.) Each such y may have several traces associated with it for the sake of different nodes devoted to infimum requirements. Suppose $z = \langle \#\alpha, x, y, w\rangle$ is a trace assigned to $\langle x, y\rangle$ for the sake of some node α devoted to an infimum requirement Q_Φ.

Let t be the greatest α-expansionary stage less than $s + 1$ (there must be such a stage since the stage at which z was assigned as a trace is α-expansionary). Check whether any number less than $\varphi(x)$ has entered A since the beginning of substage $|\alpha|$ of stage t (that is, since the exact substage at which the largest length of agreement between Φ^A and Φ^C was established). The idea is that unless such a change has occurred in A, there is no need to notify the infimum since A has held its side of the

computation on x. If such a change in A has indeed occurred, then enumerate the trace z into both A and B, and reset all coding or infimum nodes $\beta \geq \alpha^\frown 1$.

2.1.2 Constructing f_{s+1} Now we construct the current approximation f_{s+1} to the true path, taking appropriate action at each node visited along the way. Perform the following substage in increasing order of t for each t with $0 \leq t \leq s$.

2.1.3 Substage t Let $\alpha = f_{s+1} \upharpoonright t$, and consider the three possibilities for the type of requirement to which α is devoted.

Case 1: If α is devoted to an infimum requirement Q_Φ, then let

$$l(\alpha, s+1) = \max\{x : (\forall y < x)[\Phi_{s+1}^{C_{s+1}}(y) \downarrow = \Phi_{s+1}^{A^+}(y)]\},$$

and let

$$m(\alpha, s+1) = \max\{l(\alpha, r) : r < s+1 \text{ and } r \in S^\alpha\}.$$

Call $s+1$ *α-expansionary* if $l(\alpha, s+1) > m(\alpha, s+1)$. Put $f_{s+1}(t) = 0$ if $s+1$ is α-expansionary, and $f_{s+1}(t) = 1$ otherwise.

If $s+1$ is α-expansionary then for each pair $\langle x, y\rangle$ with $x < l(\alpha, s+1)$ and $y < \varphi(x)$ which does not already have an uncancelled α-trace, find the least number w such that $z = \langle \#\alpha, x, y, w\rangle > s+1$ and assign z as an α-trace for $\langle x, y\rangle$. Finally if $s+1$ is α-expansionary then reset every coding or infimum node $\beta \geq \alpha^\frown 1$.

Case 2: If α is devoted to a coding requirement P_Φ, we automatically have $f_{s+1}(t) = 0$. Let

$$l(\alpha, s+1) = \max\{x : (\forall y < x)[\Phi_{s+1}^{C_{s+1}}(y) = A^+(y)]\}.$$

Let r be the greatest stage less than $s+1$ at which α was reset, and enumerate into A all elements of the set $\{\langle \#\alpha, r, k\rangle : k \in K_{s+1}\}$ which do not exceed $l(\alpha, s+1)$. If the set of numbers so enumerated is nonempty, then reset every coding or infimum node $\beta > \alpha$.

Case 3: Finally, if α is devoted to a preservation requirement N_Φ, we automatically have $f_{s+1}(t) = 0$. Let

$$l(\alpha, s+1) = \max\{x : (\forall y < x)[\Phi_{s+1}^{A^+} = C_{s+1}(y)]\}$$

and

$$m(\alpha, s+1) = \max\{l(\alpha, r) : r < s+1 \text{ and } r \in S^\alpha\}.$$

If $l(\alpha, s+1) > m(\alpha, s+1)$, call $s+1$ *α-expansionary*. If $s+1$ is α-expansionary then reset every coding or infimum node $\beta > \alpha$. This completes the construction.

As remarked earlier, both reductions $B \leq_{\mathrm{w}} A$ and $B \leq_{\mathrm{w}} C$ should be clear. The existence of the true path f should also be clear. So it only remains to show that each node on the true path satisfies the requirement to which it is assigned. This is accomplished by an inductive lemma.

Lemma 2.2 *For each w-reduction Φ,*

(i) P_Φ is satisfied, and the node on the true path devoted to P_Φ causes finitely many elements to enter A and resets other nodes finitely often,

(ii) *N_Φ is satisfied and the node on the true path devoted to N_Φ resets other nodes finitely often, and*

(iii) *Q_Φ is satisfied.*

Proof: (i) Let α denote the node on the true path devoted to P_Φ. Note that by the inductive hypothesis, α is reset finitely often by preservation nodes and other coding nodes. Since infimum nodes above α are either finitary or do not reset α, α is in fact reset only finitely often. Let r be the largest stage at which α is reset. Assume for a contradiction that $A = \Phi^C$. Then $l(\alpha, s) \longrightarrow \infty$, so eventually all numbers of the form $\langle \#\alpha, r, k\rangle$ with $k \in K$ enter A. In fact we have $k \in K$ if and only if $\langle \#\alpha, r, k\rangle \in A$, so $K \leq_1 A \leq_{\mathrm{w}} C$, a contradiction.

Since P_Φ is satisfied, $q = \max\{l(\alpha, s) : s \in \omega\}$ is finite, and α never enumerates any number larger than q into A. Since α resets nodes only at stages when it enumerates into A, this can happen only finitely often as well.

(ii) Let β denote the node on the true path devoted to N_Φ. Suppose that $\Phi^A = C$. Then $l(\beta, s) \longrightarrow \infty$. We will show for a contradiction that C is recursive. Choose a stage s large enough so that after stage s, no node to the left of β, no finitary infimum node above β, and (by inductive hypothesis) no coding node above β enumerates any numbers into A. To compute $C(x)$ for some x, let s_x be the least β-expansionary stage greater than s with $l(\beta, s_x) > x$. Then we claim that $A \upharpoonright \varphi(x) = A_{s_x} \upharpoonright \varphi(x)$, so $C(x) = \Phi^A(x) = \Phi_{s_x}^{A_{s_x}}(x)$. For suppose that some number less than $\varphi(x)$ enters A after stage s_x. Let z be the smallest number to do so. By choice of s_x, z could only enter A for the sake of an infimum node $\sigma \subset \beta$ for which there are infinitely many σ-expansionary stages. In particular this means that s_x is itself σ-expansionary. But then z could only enter A after s_x if a smaller number enters first, contradicting the choice of z.

Since N_Φ is satisfied, there are only finitely many β-expansionary stages, so β resets other nodes finitely often.

(iii) Let γ denote the node on the true path devoted to Q_Φ. As in part (i), it should be clear that γ is reset only finitely often. Let r be the greatest stage at which γ is reset, and assume that the hypothesis of Q_Φ holds, namely that $\Phi^A = \Phi^C = D$. To compute $D(x)$ from B, let s_x be the least γ-expansionary stage greater than r for which $l(\gamma, s_x) > x$, $B_{s_x} \upharpoonright \varphi(x) = B \upharpoonright \varphi(x)$, and $B_{s_x}(z) = B(z)$ for every γ-trace associated with x.

Let p denote the common value of $\Phi_{s_x}^{C_{s_x}}(x)$ and $\Phi_{s_x}^{A^+}(x)$ at the end of the γ-expansionary substage. We claim that for every stage $t \geq s_x$, either $\Phi_t^{A_t}(x) = p$ or $\Phi_t^{C_t}(x) = p$, so $D(x) = p$. For since B has settled down below $\varphi(x)$, only coding nodes could cause an injury to the computation from A, and in particular only coding nodes $\tau \supset \gamma$, since all other coding nodes have either stopped acting or have been reset by γ. But if such a node (or nodes) enumerates numbers less than $\varphi(x)$ into A, then C must remain unchanged below $\varphi(x)$ until at least the next γ-expansionary substage. Otherwise, a γ-trace for x would enter B, contrary to the choice of s_x. Thus, between γ-expansionary substages, either $A \upharpoonright \varphi(x)$ or $C \upharpoonright \varphi(x)$ is preserved, as desired. This completes the proof of the lemma, and hence of the theorem. □

3 Noncapping In contrast to Theorem 2.1 we now show that every r.e. w-degree is half of a pair without infimum.

Theorem 3.1 *For any nonrecursive, w-incomplete r.e. w-degree* $\mathbf{c}$*, there is an r.e. w-degree* $\mathbf{a}$ *such that the infimum* $\mathbf{a} \cap \mathbf{c}$ *fails to exist.*

Proof: Let C be an r.e. set of the given degree $\mathbf{c}$ with recursive enumeration $\{C_s\}_{s\in\omega}$. The proof will be nonuniform in the sense that we will construct two r.e. sets A and $\hat{A}$, only one of whose w-degrees is guaranteed to have the desired property.

Our requirements are therefore based on pairs of sets. In addition to constructing the main sets A and $\hat{A}$, we will construct for each quadruple $(V, \hat{V}, \Psi, \hat{\Psi})$, where V and $\hat{V}$ are r.e. sets and Ψ and $\hat{\Psi}$ are w-reductions, a corresponding pair B and $\hat{B}$. The idea is that if $\Psi^A = \Psi^C = V$ and $\hat{\Psi}^{\hat{A}} = \hat{\Psi}^C = \hat{V}$, then either B will be w-reducible to both A and C but not V (so that $\deg_{\mathrm{w}}(V) \neq \deg_{\mathrm{w}}(A) \cap \deg_{\mathrm{w}}(C)$), or $\hat{B}$ will have the corresponding relationship with $\hat{A}$, C, and $\hat{V}$.

If we can do this for each such quadruple then we are done. For if the w-degree of V is actually the infimum of those of A and C (for example), then we will have guaranteed that no $\hat{V}$ could have a w-degree which is the infimum of those of C and $\hat{A}$.

3.1 Requirements There are three types of requirements we will satisfy in order to achieve the desired results. First, for each pair V and $\hat{V}$, and each pair of w-reductions Ψ and $\hat{\Psi}$, we have a requirement

$$P_{V,\hat{V},\Psi,\hat{\Psi}} : \Psi^A = \Psi^C = V \ \&\ \hat{\Psi}^{\hat{A}} = \hat{\Psi}^C = \hat{V} \longrightarrow B \leq_{\mathrm{w}} A \ \&\ B \leq_{\mathrm{w}} C.$$

To facilitate the notation, we will suppress the subscripts and call this a P-type requirement. For each P-type requirement, and each w-reduction Φ, we have a Q-type subrequirement

$$Q : \Psi^A = \Psi^C = V \ \&\ \hat{\Psi}^{\hat{A}} = \hat{\Psi}^C = \hat{V} \ \&\ \Phi^V = B \longrightarrow \hat{B} \leq_{\mathrm{w}} \hat{A} \ \&\ \hat{B} \leq_{\mathrm{w}} C.$$

Finally for each Q-requirement and each w-reduction $\hat{\Phi}$ we have an R-type subrequirement

$$R : \Psi^A = \Psi^C = V \ \&\ \hat{\Psi}^{\hat{A}} = \hat{\Psi}^C = \hat{V} \ \&\ \Phi^V = B \longrightarrow \hat{\Phi}^{\hat{V}} \neq \hat{B}.$$

3.2 Strategy Consider how we might try to satisfy a single triple of P-,Q-, and R-type requirements. We need take no action at all until we see a certain amount of agreement between Ψ^A and Ψ^C and between $\hat{\Psi}^{\hat{A}}$ and $\hat{\Psi}^C$. As this agreement grows, we must ensure that the conclusion of the P-requirement holds, which we will do by constructing a functional Δ which will give the desired reductions $B = \Delta^A = \Delta^C$. Similarly, we need only take action on behalf of the Q-requirement if we begin to see additional agreement between Φ^V and B, in which case we will extend the definition of another functional $\hat{\Delta}$ which will satisfy $\hat{B} = \hat{\Delta}^{\hat{A}} = \hat{\Delta}^C$.

Since Q and R have the same hypotheses, we must also take action to satisfy R. To this end, we begin execution of the following algorithm.

1. We pick a number $\hat{x}$ for possible enumeration into both $\hat{B}$ and $\hat{A}$.
2. Wait until $\hat{\Phi}^{\hat{V}}(\hat{x}) \downarrow = 0$ and $\hat{\Psi}^{\hat{A}} \upharpoonright \hat{\varphi}(\hat{x}) = \hat{\Psi}^{C} \upharpoonright \hat{\varphi}(\hat{x}) = V \upharpoonright \hat{\varphi}(\hat{x})$. (If this never happens, then R is satisfied.) When this occurs, restrain numbers less than $\max\{\hat{\psi}(y) : y \leq \hat{\varphi}(\hat{x})\}$ from entering $\hat{A}$, so that $\hat{A}$ indirectly maintains the 0 computation from $\hat{V}$.
3. Wait for C to change below $\hat{x}$. While waiting, repeat steps 1 and 2. If we keep passing step 2, then eventually some such C change must occur, since C is non-recursive.
4. Wait for $\hat{\Psi}^{C}$ to recompute $\hat{V} \upharpoonright \hat{\varphi}(\hat{x})$.

At this point, we have C permission to enumerate $\hat{x}$ into $\hat{B}$, and C is maintaining the 0 computation from $\hat{V}$. So we would like to enumerate $\hat{x}$ into both $\hat{B}$ and $\hat{A}$, redefine $\hat{\Delta}(\hat{x})$, and wait for $\hat{A}$ to reestablish its control over $\hat{V}$. The problem is that C might change before this happens. So we hold off on enumerating, and instead start working on the unhatted side to set up a situation where a C change would be beneficial.

5. For each n starting with $n = 0$, perform the following subroutine.
   (a) Pick a number x_n for possible enumeration into both B and A.
   (b) Wait until $\Phi^{V}(x_n) \downarrow = 0$ and $\Psi^{A} \upharpoonright \varphi(x_n) = \Psi^{C} \upharpoonright \varphi(x_n) = V \upharpoonright \varphi(x_n)$. (If this never happens then the hypothesis of Q is falsified.) When this occurs, restrain numbers less than $m = \max\{\psi(y) : y \leq \varphi(x_n)\}$ from entering A, so that A indirectly maintains the 0 computation from V.
   (c) Wait for n to enter K. The idea here is that since C is w-incomplete, if there are infinitely many x_n, then for infinitely many of them, n will enter K after C has already settled down below x_n (else we could compute K from C). Thus by waiting for n to enter K before committing ourselves to using x_n as a β-witness, we are using the w-incompleteness of C as a pseudo-restraint on C. While waiting, repeat 5(a) and 5(b) for the next value of n.
   (d) Put x_n into A.
   (e) Wait for Ψ^{A} to recompute $V \upharpoonright \varphi(x)$. If C changes below m before this happens, then abandon x_n and start over at 5(a) with the next value of n. Since C is w-incomplete, we will eventually get a recomputation before any such C change.

Now the unhatted side is just waiting for a C permission, so we return to the hatted side as previously planned.

6. Put $\hat{x}$ into both $\hat{A}$ and $\hat{B}$.
7. Wait for $\hat{\Psi}^{\hat{A}}$ to recompute $\hat{V} \upharpoonright \hat{\varphi}(\hat{x})$. If C changes below x_n before this happens, then we can enumerate x_n into B to falsify the hypothesis of requirement Q. Otherwise we have at least satisfied requirement R.

3.3 The priority tree In order to satisfy all of the requirements simultaneously, we use a priority tree, with each node devoted to a single requirement. Nodes devoted to P-, Q-, and R-type requirements are called α, β, and γ nodes, respectively. Each

α node has its own functional Δ, and each β node has its own functional $\hat{\Delta}$ as described in the general strategy above. We regard both α and β nodes as having two possible outcomes, depending on whether or not the hypotheses of their requirements appear infinitely often to be satisfied. However, γ nodes will have a single outcome (success). Thus our tree is a subtree of the binary tree $2^{<\omega}$.

More precisely, we can define the tree T recursively as follows. Fix a priority ordering of the set of all P-, Q-, and R-type requirements such that any subrequirement Q comes after its associated P-type requirement and any subrequirement R comes after its associated Q-type requirement.

For any node $\sigma \in T$, let S be the highest priority requirement such that

1. S is not yet assigned to any node $\tau \subset \sigma$;
2. if S is a Q-type requirement, then $\alpha^\frown 0 \subseteq \sigma$ where α is devoted to S's associated P-type requirement.
3. if S is an R-type requirement, then $\beta^\frown 0 \subseteq \sigma$ where β is devoted to S's associated Q-type requirement.

Then σ will be devoted to S, and will have two successors $\sigma^\frown 0$ and $\sigma^\frown 1$ unless S is an R-type requirement, in which case it has the single successor $\sigma^\frown 1$.

At each stage s of the construction, we will construct a string $f_s \in T$ of length at most s, which will be our current approximation to the true path, that is, the path $f \in [T]$ for which $f \upharpoonright m = \liminf_s f_s \upharpoonright m$ for all m, in the sense that for $\beta = f \upharpoonright m$ we have

(a) $(\exists^{<\infty} s)[f_s <_L \beta]$ and

(b) $(\exists^{\infty} s)[\beta \subseteq f_s]$.

Given a node $\tau \in T$, a stage s is called a τ-*stage* if $\tau \subseteq f_s$. The set of all τ-stages is denoted by S^τ.

3.4 The construction We build all sets and functionals in stages. At a given stage s, we construct the current approximation f_s to the true path, taking action for each of the nodes visited along the way. An α-node will always be either *active* for some lower β, meaning it is ready to perform step 7 of the algorithm, or *passive*, meaning that if visited, it should perform its default action of extending the definition of Δ.

Similarly, a β node will be marked as active for some lower γ if it has started performing step 5 of the algorithm, and passive otherwise, in which case it should simply extend the definition of $\hat{\Delta}$. During the construction below, to reset an α node means to mark α as passive, abandon the current functional Δ associated with α and start building a new functional (which we will still denote by Δ to ease the notation). To reset a β node means to mark β as passive, cancel any uncancelled β-witnesses, and start over with a new functional $\hat{\Delta}$. To reset a γ node means to cancel any uncancelled γ-witnesses.

Stage $s+1$: We define f_{s+1} recursively for arguments $n < s+1$. Assume that $f_{s+1} \upharpoonright n$ is defined. Depending on whether $f_{s+1} \upharpoonright n$ is an α, β, or γ node, we take action accordingly and define $f_{s+1}(n)$.

Case 1 ($f_{s+1} \upharpoonright n$ is an α node): Let $l(\alpha, s+1)$ denote the length of agreement

$$l(\alpha, s+1) = \max\{y : \forall x < y[\Psi^A(x) = \Psi^C(x) = V(x) \ \& \ \hat{\Psi}^{\hat{A}}(x) = \hat{\Psi}^C(x) = \hat{V}(x)]\},$$

and let $m(\alpha, s+1) = \max\{l(\alpha, t) : t < s+1 \ \& \ t \in S^\alpha\}$. Stage $s+1$ is called α-expansionary if $l(\alpha, s+1) > m(\alpha, s+1)$.

If $s+1$ is not α-expansionary, then just define $f_{s+1}(n) = 1$ and take no action for α. If $s+1$ is α-expansionary, then define $f_{s+1}(n) = 0$ and take the following action. If α is active for some $\beta \supset \alpha$, then β must have a unique uncancelled β-witness $x_n \in A$ and must in turn be active for some $\gamma \supset \beta$ that has an uncancelled γ-witness $\hat{x} \in \hat{A}$. Check if $\hat{\Delta}^C(\hat{x})$ is still defined. If not, then put x_n into B. Define $\Delta^A(z) = \Delta^C(z) = B(z)$ for all $z < l(\alpha, s+1)$, and return α to passive mode.

If α is passive, then check whether there is any $\beta \supset \alpha$ that is active for some $\gamma \supset \beta$ and has an uncancelled β-witness $x_n \in A$ (i.e., β has performed step 5(d) of the algorithm). If so, then check whether $\Delta^C(x_n)$ is still defined. If so, then put $\hat{x}$ into $\hat{A}$ and $\hat{B}$ but do not redefine Δ. Mark α as active for β, stop building f_{s+1}, and proceed directly to the next stage. If $\Delta^C(x_n)$ is no longer defined, then cancel x_n, define $\Delta^A(z) = \Delta^C(z) = B(z)$ for all $z < l(\alpha, s+1)$, and leave α in passive mode.

Finally if α is passive but there is no such β as in the previous paragraph, then just define $\Delta^A(z) = \Delta^C(z) = B(z)$ for all $z < l(\alpha, s+1)$, and leave α in passive mode.

Case 2 ($f_{s+1} \upharpoonright n$ is a β node): Define the length of agreement $l(\beta, s+1)$ by $l(\beta, s+1) = \min\{l(\alpha, s+1), \bar{l}(\beta, s+1)\}$, where α is the α node associated with β and

$$\bar{l}(\beta, s+1) = \max\{y : \forall x < y[\Phi^V(y) = B(y) \ \& \ \Psi^C \upharpoonright \varphi(y) = \Psi^A \upharpoonright \varphi(y) = V \upharpoonright \varphi(y)]\}.$$

Let $m(\beta, s+1) = \max\{l(\beta, t) : t < s+1 \ \& \ t \in S^\beta\}$ and call stage $s+1$ β-expansionary if $l(\beta, s+1) > m(\beta, s+1)$.

If $s+1$ is not β-expansionary, then just define $f_{s+1}(n) = 1$ and take no action for β. If $s+1$ is β-expansionary, then define $f_{s+1}(n) = 0$ and take the following action. If β is active for some $\gamma \supset \beta$, first check if there is any uncancelled γ-witness $\hat{x} \in \hat{A}$. If so, then we must have passed step 6 of the algorithm, and the fact that we are visiting β again means that in fact we have performed step 7 as well. Define $\hat{\Delta}^{\hat{A}}(z) = \hat{\Delta}^C(z) = \hat{B}(z)$ for all $z < l(\beta, s+1)$, and return β to passive mode.

If there is no such γ-witness, check whether β has an uncancelled witness x_n for γ with $\Phi^V(x_n) \downarrow = 0$, x_n not yet in A, and $n \in K$. If so, enumerate x_n into A. Stop building f_{s+1}, and proceed directly to the next stage. If there is no such γ- or β-witness as above, check whether $\Phi^V(x_n) \downarrow = 0$ for every uncancelled β witness x_n. If so then pick the least n such that there is no β-witness x_n for γ, and assign the least element $x_n \in \omega^{[\langle \alpha, \beta \rangle]}$ greater than $s+1$ as a new β-witness for γ. Stop building f_{s+1}, and proceed directly to the next stage.

If, on the other hand, β has an uncancelled witness x_n for γ with $\Phi^V(x_n) \neq 0$, then simply continue to wait for $\Phi^V(x_n) \downarrow = 0$. Stop building f_{s+1}, and proceed directly to the next stage. If β is passive, then first check whether there is any γ node

$\gamma \supseteq \beta\hat{}0$ that has no uncancelled γ-witness in A but has an uncancelled, realized γ-witness $\hat{x}$ for which $C_{s+1} \upharpoonright \hat{x} \neq C_t \upharpoonright \hat{x}$, where t is the stage at which $\hat{x}$ became realized. If there is any such node, let γ denote the one of highest priority, and mark β as active for γ. Stop building f_{s+1}, and proceed directly to the next stage. If there is no such γ node, then define $\hat{\Delta}^{\hat{A}}(z) = \hat{\Delta}^{C}(z) = \hat{B}(z)$ for all $z < l(\beta, s+1)$, and define $f_{s+1}(n) = 0$.

Case 3 ($f_{s+1} \upharpoonright n$ is a γ node): We must define $f_{s+1}(n) = 1$ in all cases. Define the length of agreement $l(\gamma, s+1)$ by $l(\gamma, s+1) = \min\{l(\beta, s+1), \bar{l}(\gamma, s+1)\}$, where β is the β node associated with γ and

$$\bar{l}(\gamma, s+1) = \max\{y : \forall x < y[\hat{\Phi}^{\hat{V}}(y) = \hat{B}(y) \ \& \ \hat{\Psi}^{C} \upharpoonright \hat{\varphi}(y) = \hat{\Psi}^{\hat{A}} \upharpoonright \hat{\varphi}(y) = \hat{V} \upharpoonright \hat{\varphi}(y)]\}.$$

Let $m(\gamma, s+1) = \max\{l(\gamma, t) : t < s+1 \ \& \ t \in S^{\gamma}\}$ and call stage $s+1$ γ-expansionary if $l(\gamma, s+1) > m(\gamma, s+1)$. If $s+1$ is not γ-expansionary, then take no action for γ. If $s+1$ is γ-expansionary, check if there is any uncancelled, unrealized γ-witness $\hat{x}$ for which $\hat{\Phi}^{\hat{V}}(\hat{x}) \downarrow = 0$. If so, pick the least such $\hat{x}$ and call it realized (and cancel the others). If there is no γ-witness $\hat{x}$ for which $\hat{\Phi}^{\hat{V}}(\hat{x}) \uparrow$, then assign the least element $x_n \in \omega^{[\alpha]}$ greater than $s+1$ as a new γ-witness.

This concludes the description of the cases. At the end of stage s+1, we reset all nodes $\tau > f_{s+1}$.

3.5 The verification To show that the construction works, first note that there is a unique path $f \in [T]$, called the true path, for which $f \upharpoonright n = \liminf_s f_s \upharpoonright n$ for all n, in the usual sense that for $\tau = f \upharpoonright m$ we have

(a) $(\exists^{<\infty} s)[f_s <_L \tau]$ and
(b) $(\exists^{\infty} s)[\tau \subseteq f_s]$.

Note also that any node $\tau \subset f$ is reset at most finitely often. It remains to show that each node along the true path f fulfills its commitment by satisfying its associated requirement.

Lemma 3.2 *If for any α node there are infinitely many α-expansionary stages, then there are infinitely many stages at which α is passive.*

Proof: If α becomes active for some $\beta \supset \alpha$, then α will become passive by the next α-expansionary stage. □

Lemma 3.3 *If for any β node there are infinitely many β-expansionary stages, then there are infinitely many stages at which β is passive.*

Proof: If β becomes active for some $\gamma \supset \beta$, then every β-witness x_n for γ will eventually satisfy $\Phi^{V}(x_n) \downarrow = 0$ since there are infinitely many β-expansionary stages. There are infinitely many n that enter K after C has settled down below x_n. Thus some such witness x_n must eventually get enumerated into A. The α node associated with β must then go active for β, and by the previous lemma must become passive

again. If β has not been reset (and so become passive) by the next β-expansionary stage, then β will become passive at that stage. □

Lemma 3.4 *Any α node $\alpha \in f$ satisfies its associated P-type requirement.*

Proof: Suppose that α is devoted to the requirement

$$\Psi^A = \Psi^C = V \ \ \& \ \ \hat{\Psi}^{\hat{A}} = \hat{\Psi}^C = \hat{V} \longrightarrow B \leq_{\mathrm{w}} A \ \ \& \ \ B \leq_{\mathrm{w}} C,$$

and that the hypothesis of this requirement is true. Then there are infinitely many α-expansionary stages, so by Lemma 3.2, there are infinitely many stages at which α is passive. If α becomes active for some $\beta \supset \alpha$ at some stage, then Δ^A and Δ^C will not be extended at that stage, but will be extended at the next stage when α becomes passive. Thus Δ^A and Δ^C are extended infinitely often. For any argument x, we always define $\Delta^A(x) = \Delta^C(x) = B(x)$, and this value only changes (from 0 to 1) if both x has entered A and $C \upharpoonright x$ has changed since the last time $\Delta^A(x)$ and $\Delta^C(x)$ were defined. So Δ gives the desired w-reductions, with the use function δ being the identity function. □

Lemma 3.5 *Any β node $\beta \in f$ satisfies its associated Q type requirement.*

Proof: Exactly as the proof of the previous lemma. □

Lemma 3.6 *Any γ node $\gamma \in f$ satisfies its associated R type requirement.*

Proof: Wait for a stage after which γ is never reset. Assuming that the hypotheses of R hold, any γ-witness that gets assigned afterward will eventually become realized. Because C is nonrecursive, the associated β node β will become active for γ, and by Lemma 3.3 must eventually become passive again. Because C is w-incomplete, the associated α node α will eventually go active for β. Now if C has changed below the witness $\hat{x}$ which caused β to go active, then we will enumerate x_n into B, after which we will have $B(x_n) = 1$ but $\Phi^{\Psi^A}(x_n) \downarrow = 0$ ever afterward, contradicting the assumption that there are infinitely many β-expansionary stages. Thus we still have $\hat{\Phi}^{\hat{\Psi}^{\hat{A}}}(\hat{x}) \downarrow = 0$, but $\hat{B}(\hat{x}) = 1$. Thus R is satisfied. □

In the r.e. Turing degrees, there are several ways to show that there exists a pair with no infimum (and hence that the r.e. Turing degrees do not form a lattice). The first proofs of this fact were given by Lachlan [10], p. 569, and Yates [14]. Lachlan's proof was based on his "non-diamond" theorem and the Sacks splitting theorem, while Yates indicated a proof by relativizing the minimal pair construction to a certain uniformly ascending sequence of r.e. degrees. Later, Jockusch [9] gave an elegant finite injury construction of a pair of r.e. Turing degrees with no infimum. Fischer [7] showed that this last construction carries over to the r.e. w-degrees to give a pair of r.e. w-degrees with no infimum. Blaylock [4] used the same non-infimum strategy combined with standard techniques to give uniform proofs of Theorem 3.1 in the case when the given degree is either low or promptly simple.

Acknowledgments Downey wishes to acknowledge the support of the Marsden Fund for Basic Science. Lempp wishes to acknowledge support by the National Science Foundation. Both also wish to acknowledge support by a U.S./New Zealand binational grant.

REFERENCES

[1] Ambos-Spies, K., "Contiguous r.e. degrees," pp. 1–37 in *Computation and Proof Theory, Lecture Notes in Mathematics*, 1104, edited by M. M. Richter et al., Springer-Verlag, New York, 1984. Zbl 0562.03022 MR 86f:03065 1

[2] Ambos-Spies, K., "On pairs of recursively enumerable degrees," *Transactions of the American Mathematical Society*, vol. 283 (1984), pp. 507–31. Zbl 0541.03023 MR 85d:03083 1

[3] Ambos-Spies, K., "Cupping and noncapping in the r.e. weak truth table and Turing degrees," *Archiv für mathematische Logik und Grundlagenforschung*, vol. 25 (1985), pp. 109–26. Zbl 0619.03032 MR 87j:03058 1

[4] Blaylock, R., *Some Results on e-Genericity and Recursively Enumerable Weak Truth Table Degrees*, Ph.D. Dissertation, University of Illinois at Urbana-Champaign, 1991. 3

[5] Cohen, P. F., *Weak Truth-Table Reducibility and the Pointwise Ordering of 1-1 Recursive Functions*, Ph.D. Dissertation, University of Illinois at Urbana-Champaign, 1975. 1

[6] Downey, R. G., "Δ_2^0 degrees and transfer theorems," *Illinois Journal of Mathematics,* vol. 31 (1987), pp. 419–27. Zbl 0629.03017 MR 89c:03070 1

[7] Fischer, P., "Pairs without infimum in the recursively enumerable weak truth table degrees," *The Journal of Symbolic Logic*, vol. 51 (1986), pp. 117–29. Zbl 0587.03030 MR 87g:03044 1, 3

[8] Friedberg, R. M., and H. Rogers, Jr. "Reducibility and completeness for sets of integers," *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, vol. 5 (1959), pp. 117–25. Zbl 0108.00602 MR 22:3682 1

[9] Jockusch, Jr., C. G., "Three easy constructions of recursively enumerable sets," pp. 83–91 in *Logic Year 1979–80, Lecture Notes in Mathematics*, 859, edited by M. Lerman, J. Schmerl, and R. Soare, Springer-Verlag, New York, 1981. Zbl 0472.03031 MR 83a:03036 1, 3

[10] Lachlan, A. H., "Lower bounds for pairs of recursively enumerable degrees," *Proceedings of the London Mathematical Society*, vol. 16 (1966), pp. 537–69. Zbl 0156.00907 MR 34:4126 3

[11] Ladner, R. E., and L. P. Sasso, "The weak truth table degrees of recursively enumerable sets," *Annals of Mathematical Logic*, vol. 8 (1975), pp. 429–48. Zbl 0324.02028 MR 52:63 1, 1

[12] Soare, R. I., *Recursively Enumerable Sets and Degrees*, Springer-Verlag, New York, 1987. Zbl 0623.03042 MR 88m:03003 1, 1

[13] Stob, M., "wtt-degrees and T-degrees of recursively enumerable sets," *The Journal of Symbolic Logic,* vol. 48 (1983), pp. 921–30. 1

[14] Yates, C. E. M., "A minimal pair of recursively enumerable degrees," *The Journal of Symbolic Logic,* vol. 32 (1965), pp. 159–68. Zbl 0143.25402 MR 34:5677 3

Individual, Inc.
8 New England Executive Parkway West
Burlington, MA 01803
email: blaylock@individual.com

Department of Mathematics
Victoria University of Wellington
P.O. Box 600
Wellington
NEW ZEALAND
email: rod.downey@vuw.ac.nz

Department of Mathematics
University of Wisconsin
480 Lincoln Drive
Madison, WI 53706-1388
email: lempp@math.wisc.edu