DEGREE THEORETIC DEFINITIONS OF THE LOW$_2$ RECURSIVELY ENUMERABLE SETS

Rod Downey
Mathematics Department
Victoria University of Wellington
P.O. Box 600
Wellington, New Zealand

Richard A. Shore
Mathematics Department
White Hall
Cornell University
Ithaca, N.Y. 14853 U.S.A.

1. Introduction.

The primary relation studied in recursion theory is that of relative complexity: A set or function A (of natural numbers) is reducible to one B if, given access to information about B, we can compute A. The primary reducibility is that of Turing, $A \leq_T B$, where arbitrary (Turing) machines, φ_e, can be used; access to information about (the *oracle*) B is unlimited and the lengths of computations are potentially unbounded. Many other interesting reducibilities result from restricitng one or more of these facets of the procedure. Thus, for example, the strongest notion considered is one–one reducibility on sets: $A \leq_1 B$ iff there is a one–one recursive (= effective) function f such that $x \in A \Leftrightarrow f(x) \in B$. Many–one ($\leq_m$) reducibility simply allows f to be many–one. Other intermediate reducibilities include truth–table ($\leq_{tt}$) and weak truth–table ($\leq_{wtt}$). The latter imposes a recursive bound $f(x)$ on the information about B that can be used to compute $A(x)$. The former also bounds the length of computations by requiring that the computation of $A(x)$ from B halt in at most $f(x)$ many steps.

Each such reducibility r defines a notion of degree, $deg_r(A) = \{B : A \leq_r B \wedge B \leq_r A\}$, and a corresponding structure $\mathcal{D}_r$ of the r–degrees ordered by r–reducibility. (We typically denote the degree of A by $\mathbf{a}$.) A major theme in recursion theory has been the investigation of the relation between a set's place in these orderings (the *algebraic* properties of its degree) and other algorithmic, set–theoretic or

Research partially supported by the NSF via Grants DMS–8912797, DMS–9204308; a U.S.-New Zealand Binational Cooperative Research Grant including NSF INT 87–22887 and INT 90–20558; by ARO through MSI, Cornell University via contract DAAL–03–C–0027; and IGC of Victoria University, Wellington.

definability type notions of complexity. Important examples of such other notions include rates of growth of functions, the types of approximation procedures which converge to the given function or set and the (syntactic) complexity of defining the set (or function) in arithmetic or analysis. A fundamental role in measuring such apparently external complexity notions is played by the *jump operator* :

$$A' = \{\langle x, y\rangle \mid \text{the } x^{th} \text{ Turing machine running on input } y \text{ with access to information about the oracle } A \text{ halts}\}.$$

We denote the n^{th} iteration of the jump by $A^{(n)}$: $A^{(0)} = A$, $A^{(1)} = A'$, $A^{(n+1)} = A^{(n)'}$. This operator relates many notions of complexity. A' is the complete set which can be enumerated recursively (with A as an oracle), i.e., if B can be so enumerated (B *is r. e. in* A) then $B \leq_1 A'$ and A' itself is r. e. in A. (Note that B being r. e. in A is the same as B being approximable recursively in A via a scheme which starts giving $B(x) = 0$ for every x and changes its mind at most once.) A classic theorem of Kleene and Post connects the jump operator to both enumerability and definability. We say A is $\Sigma_n(\Pi_n)$ in B if $y \in A$ is definable from B by a formula of arithmetic of the form $\exists x_1 \forall x_2 \cdots Q x_n S(\overrightarrow{x}, y, B)$ $(\forall x_1 \exists x_2 \ldots \overline{Q} x_n S(\overrightarrow{x}, y, B)$ where S has only bounded quantifiers (or equivalently where S is a recursive relation). A is Δ_n in B if it is both Σ_n and Π_n in B.

Theorem. *(Kleene and Post)*
i) A is Σ_{n+1} in B iff A is r. e. in $B^{(n)}$ iff $A \leq_m B^{(n+1)}$ iff $A \leq_1 B^{(n+1)}$.
ii) A is Δ_{n+1} in B iff $A \leq_T B^{(n)}$.

(A proof can be found in Soare [1987, IV]. We refer to this text for all unexplained notions and notations.)

There are many important connections between the jump operator and rates of growth properties. Some of them will play a crucial role in this paper.

Clearly the most remarkable result relating the jump operator to the ordering of degrees is Cooper's recent theorem [1993] that the jump operator is definable, from ordering alone, in $\mathcal{D}_T$, the Turing degrees with $\leq_T$. Slaman and Woodin [1996] have used this result to considerably improve our knowledge about definability within, and possible automorphisms of, $\mathcal{D}_T$. One of their most remarkable results connects $\mathcal{D}_T$ with $\mathcal{R}_T$, the Turing degrees of the r. e. sets: If $\mathcal{R}_T$ is rigid (i.e., has no nontrivial automorphisms), then so is $\mathcal{D}_T$.

Although much is known about the global structure of the degrees as a whole under many reducibilities (see for example Nerode and Shore [1980 a,b] and Shore [1985] as well as Slaman and Woodin [1996]), almost nothing in this vein is known about the r. e. degrees. We believe that, as in the degrees as a whole, a key problem for the study of the r. e. sets and degrees should be the elucidation of the relation between their degree structure and the jump operator.

The standard approach to this problem is to examine the properties of the jump classes.

(1.1)Definition. *A degree* $\mathbf{a} < \mathbf{0}'$ *is high*$_n$ *iff* $\mathbf{a}^{(n)} = \mathbf{0}^{(n+1)}$ *(its* n^{th} *jump is as high as possible). The degree is low*$_n$ *iff* $\mathbf{a}^{(n)} = \mathbf{0}^{(n)}$ *(its* n^{th} *jump is as low* *as possible). We also use $\overline{high}_n$ and $\overline{low}_n$ to denote the nonhigh$_n$ and nonlow$_n$ degrees, respectively. If $n = 1$, we usually omit the subscript.*

There have been many important results connecting the lattice structure $\mathcal{E}$ of the r. e. sets and these jump classes. The first was the characterization in Martin [1966] of the *high* r. e. sets A in terms of rates of growth (domination) properties and as the degrees containing co–atoms of $\mathcal{E}^*$, the lattice $\mathcal{E}$ modulo the ideal $\mathcal{F}$ of finite sets. The other jump class that has been characterized lattice theoretically is that of the nonlow$_2$ r. e. degrees: $\mathbf{a}$ is nonlow$_2$ iff there is an $A \in \mathbf{a}$ with no maximal superset (Lachlan [1968a] and Shoenfeld [1976]).

Although important relations have been found between the jump operator and algebraic properties of r. e. degree structures, there have been no characterizations or definability results. In the former category, an important early result is Cooper [1973]: If $\mathbf{a}$ is high then there are nonrecursive $\mathbf{b}, \mathbf{c} < \mathbf{a}$ such that $\mathbf{b} \wedge \mathbf{c} = 0$. In the latter category, the best results follow from Shore and Slaman [1990] and [1993]: There are definable properties in $\mathcal{R}_T$ which separate the high r. e. degrees from the low$_2$ ones.

The most important result in this paper is that the low$_2$ degrees are definable in $\mathcal{R}_{tt}$, the structure of the r. e. truth–table degrees with $\leq_{tt}$.

Theorem 3.3. (ii) *An r. e. set A is low$_2$ if and only if its degree has a minimal cover in $\mathcal{R}_{tt}$.*

This result was suggested by A. Nies. It is proven by showing, on the one hand, that the nonlow$_2$ r. e. truth table degrees are dense (Theorem 2.7 (i)) and, on the other, that every low$_2$ r. e. degree has a minimal cover in $\mathcal{R}_{tt}$ (Theorem 3.3 (i)). The first result requires adapting techniques for dealing with nonlow$_2$ sets in the non r. e. T–degrees to r. e. sets. These techniques exploit the characterization of nonlow$_2$ sets (below $\mathbf{0}'$) in terms of domination properties. We use a full approximation procedure to adapt these techniques to the construction of r. e. sets. We have been able to use our techniques to give unified proofs of most known nonlow$_2$ arguments for r. e. sets as well as other new results. For instance, we show that if an r. e. set A has minimal *tt*–degree then A is low$_2$. This answers an old question about $\mathcal{R}_{tt}$ (see Odifreddi [1981], [1989]). In a future paper we will extend these methods to prove some new embedding results in $\mathcal{R}_T$, e.g., the basic nonmodular five element lattice 1–3–1 can be embedded in $\mathcal{R}_T$ below every nonlow$_2$ degree. (Note that Downey [1990] shows that, for every r. e. $\mathbf{a} > \mathbf{0}$, there is a nonrecursive r. e. $\mathbf{d} < \mathbf{a}$ below which 1–3–1 cannot be embedded.)

The second result (on minimal covers) introduces techniques extending those of Shore and Slaman [1990] and [1993]. We also use these techniques to prove that the low$_2$ r. e. sets can be defined using Turing reducibility and any one of the stronger reducibilities mentioned above.

Theorem 3.5. *i) If $\mathbf{c}$ is low$_2$ then there exists an incomplete 1–topped r. e. T–degree $\mathbf{a} \geq \mathbf{c}$. (We say that $\mathbf{a}$ is r–topped if there is an r. e. $A \in \mathbf{a}$ such that every r. e. B Turing below A is, in fact, below A with respect to r–reducibility.)*

As $\mathbf{c}$ being 1–topped implies it is r–topped for $r = m$, tt and wtt and each of these properties implies that $\mathbf{c}$, if incomplete, is low$_2$, this theorem shows that the incomplete r. e. T–degrees with an r. e. r–topped degree above them are precisely the low$_2$ r. e. degrees.

2. Working below a nonlow$_2$ degree.

In this section we develop a reasonably general technique for working below a nonlow$_2$ r. e. degree. In the global degrees, there is a well known technique to handle this problem. A degree $\mathbf{d} < \mathbf{0}'$ is nonlow$_2$ iff for every function $h \leq_T \emptyset'$, there is a function g recursive in $\mathbf{d}$ such that g is not dominated by h. This characterization is used for "$\overline{low}_2$ permitting" as follows: Relying on specific properties of the requirements to be met, one defines "in advance" a function h which gives an appropriate "search space" inside of which one should look for witnesses to satisfy some requirements. This function will be recursive in $\emptyset'$ due to the specific nature of the relevant requirements. Now if $\mathbf{d}$ is nonlow$_2$, there is a strictly increasing function g recursive in $\mathbf{d}$ not dominated by h. The idea then is to use g to $\mathbf{d}$–recursively bound searches and hence make the construction an oracle one recursive in $\mathbf{d}$. By the way g and h have been constructed, the fact that $g(s) > h(s)$ infinitely often guarantees that, by a priority argument, we get to meet all the requirements.

To illustrate this procedure consider the following example:

If $\mathbf{a} \in \overline{low}_2$ *then* $\exists \mathbf{b} < \mathbf{a}$ *such that* $\mathbf{b}$ *is* 1–*generic* (Jockusch–Posner [1978]).

The reader should recall that A is 1–generic if, for all r. e. sets of strings V_e, either $(\exists \sigma)(\sigma \in V_e \ \& \ \sigma \subset A)$ or $(\exists \sigma \subset A)(\forall \tau \supseteq \sigma)(\tau \notin V_e)$.

To prove the Jockusch–Posner result we build an ascending sequence of strings σ_s with the intention of setting $A = \cup \sigma_s$. (We systematically confuse a set and its characteristic function.) It clearly suffices to meet the requirements

$$R_e : \exists s[\sigma_s \in V_e \text{ or } \forall \tau \supseteq \sigma_s(\tau \notin V_e)]$$

for those V_e which are closed under extensions, i.e., $\sigma \in V_e \ \& \ \tau \supseteq \sigma \rightarrow \tau \in V_e$. We begin by defining two functions.

$$k(\tau, e) = \begin{cases} \min\{s \mid (\exists \sigma)(\tau \subseteq \sigma \& \sigma \in V_{e,s})\} \text{ if one exists} \\ 0 \text{ otherwise.} \end{cases}$$

$$h(s) = \max\{k(\tau, e) : e \leq s \& |\tau| = s\}.$$

.

Now $h \leq_T \emptyset'$ and hence there is an increasing g recursive in $\mathbf{a}$ not dominated by h. So $(\exists^\infty s)(h(s) < g(s))$. We procede as follows. Let $\sigma_0 = \lambda$. At stage $s+1$ look for the least *unsatisfied* $e \leq s$ for which there is a σ such that $\sigma_s \subseteq \sigma$, $|\sigma| \leq g(s)$ and $\sigma \in V_{e,g(s)}$. If e exists, choose the smallest such σ and let σ_{s+1} be the least string extending σ_s of length $s+1$ contained in σ. (Note that we do not set $\sigma_{s+1} = \sigma$ here, but only set it to be the initial segment of σ of length $s+1$.) Declare e to be *satisfied* if $\sigma \subseteq \sigma_{s+1}$. If no such e exists, set $\sigma_{s+1} = \sigma_s * 0$. Clearly the construction is recursive in $\mathbf{a}$. It is easy to see that as g is bigger than h infinitely often, we will meet each R_e. By induction and the choice of g we may suppose σ_s meets

R_i for $i < e$ and $g(s) > h(s)$. If $\forall \tau \supseteq \sigma_s$ $(\tau \notin V_e)$ or $\exists \tau \subseteq \sigma_s$ $(\tau \in V_e)$ we have also met R_e. If not, then $\exists \tau \supset \sigma_s$ $(\tau \in V_e)$ and indeed, by the definition of h, $\exists \tau \supseteq \sigma_s$ $(\tau \in V_{e,h(s)})$. By our choice of s, $g(s) > h(s)$ and so the next steps of the construction will successively extend σ_s until it becomes its least extension in $V_{e,g(s)}$. □

If we try to modify the ideas above to suit the r. e. degrees we face special problems. The natural idea we pursue is to use the global characterization of $\overline{\text{low}_2}$ and then approximate the functions g and h via the limit lemma. Now we will be given an r. e. set D of nonlow$_2$ degree and a "witness" function h for the satisfaction of some requirements. Again h will be recursive in $\emptyset'$. We apply the limit lemma to h so that $h(x) = \lim_s h(x,s)$ with $h(x,s)$ a recursive approximation to h. Let $D = \cup_s D_s$ be an r. e. member of the nonlow$_2$ r. e. degree $\mathbf{d}$. Again, since D is not low$_2$, there will be a function g recursive in D not dominated by h which we can also approximate via the limit lemma. Since g is recursive in D, there is a reduction $\Gamma(D) = g$. The problem is that, as $h(x)$ only equals $\lim_s h(x,s)$ and $g(x)$ only equals $\lim_s g(x,s)$, we must be able to "correct" our mistakes. This is a serious problem since the objects we need to construct must not only be recursive in D *but also r. e.* A crucial theme in our constructions below nonlow$_2$ r. e. sets will be that we can correct the mistakes that occur when $g(x,s)$ does not have its final value by dumping elements into the set we are constructing whenever $g(x,s)$ or even $D \restriction g(x,s)$ changes (we will tie $g(x,s)$ to the standard approximation of $\Gamma(D)$). To facilitate this procedure we require that our recursive approximations $h(x,s)$ and $g(x,s)$ have certain properties. In particular, since we will only be concerned with values where $g(x)$ is bigger than $h(x)$, we can always presume approximations to D, g and h so that the following hold.

Conventions:

(i) $g(x,x) > h(x,x)$,
(ii) If $g(x,s+1) \neq g(x,s)$ then $\exists z (z \in D_{s+1} - D_s$ and $z < g(x,s))$.
(iii) If $g(x,s+1) \neq g(x,s)$ then $g(x,s+1) > h(x,s)$.
(iv) $g(x,s)$ and $h(x,s)$ are monotonic in both variables.
(v) If $g(x,s) \neq g(x,s+1)$ then $g(x,s+1) = s$.
(vi) $g(x,s+1) \neq g(x,s)$ for at least one s.

We begin with some easy applications of our technique. Downey, Jockusch and Stob [1990] define a strong array $S = \{F_x : x \in \omega\}$ to be a *very strong array* (vsa) if, for all x, $|F_{x+1}| > |F_x|$ and $\cup_x \{F_x\} = \omega$. They call an r. e. set A S–anr (*array nonrecursive*) if, for all e, there is an x with $F_x \cap W_e = A \cap F_x$. Now it turns out that this notion is invariant in the following sense. If S_1 and S_2 are vsa's and $\mathbf{a}$ contains an S_1–anr set, then $\mathbf{a}$ contains an S_2–anr set. Thus we can define an anr degree to be one containing an S–anr set for some vsa S. Such sets were introduced to capture the "multiple permitting" character of several arguments, such as Martin and Pour–El [1970] and Jockusch and Soare [1972]. One of their theorems, proven indirectly in Downey et al. [1990], was that the $\overline{low_2}$ r. e. degrees are all anr. We give a simple direct proof that illustrates our ideas.

(2.1)Theorem. *(Downey et al. [1990]) If* $\mathbf{a}$ *is* $\overline{low_2}$ *and r. e. then* $\mathbf{a}$ *is anr. That is,* $\mathbf{a}$ *contains an anr r. e. set.*

Proof Let $S = \{F_x : x \in \omega\}$ be a vsa. Let D be a nonlow$_2$ r. e. set. We build an S–anr r. e. set A with $A \equiv_T D$. We need to satisfy each requirement

$$R_e : (\exists x)(F_x \cap W_e = A \cap F_x).$$

We devote $\{F_{\langle e+1,x\rangle} : x \in \omega\}$ to meeting R_e and $\{F_{\langle 0,x\rangle} : x \in \omega\}$ to coding. So we automatically put all of $F_{\langle 0,x\rangle}$ into A_s iff $x \in D_s$. Hence $A \ _T\geq D$. Let

$$h(x) = (\mu s)(\forall e \leq x)(F_{\langle e+1,x\rangle} \cap W_{e,s} = F_{\langle e+1,x\rangle} \cap W_e).$$

Then $h \leq_T \emptyset'$. So, as D is $\overline{low}_2$, there is a function $g \leq_T D$ not dominated by h, i.e., $(\exists^\infty s)(g(s) > h(s))$, with an approximation $g(x,s)$ satisfying our conventions. Now, at stage s, we compute $g(x,s)$ for all $x \leq s$. If $g(x,s) \neq g(x,s-1)$ we ensure that $F_{\langle e+1,x\rangle} \cap W_{e,g(x,s)} = F_{\langle e+1,x\rangle} \cap A_s$ for all $e \leq s$ by enumerating numbers into A as necessary. As $g(x,s)$ only changes if D changes by (ii), $A \leq_T D$. Finally, note that R_e is met. To see this, choose the least $x > e$ such that $g(x) > h(x)$. Now let s be the first stage $> x$ at which $g(x,s) = g(x)$. It follows that $F_{\langle e+1,x\rangle} \cap W_{e,h(x,s)} = A \cap F_{\langle e+1,x\rangle} = F_{\langle e+1,x\rangle} \cap W_e$, and hence R_e is met. □

The previous theorem was originally established via an index set argument. The next result is also a corollary of index set theorems (Yates [1969]). We supply a simple direct proof.

(2.2) Theorem. *If A is r. e., $\overline{low}_2$ and $A \not\equiv_{wtt} \emptyset'$ then there exists an r. e. $B \equiv_T A$ such that $B \not\leq_{wtt} A$.*

Proof: We are given A and construct B to meet each requirement

$$R_e : \Delta_e(A) \neq B,$$

where Δ_e is the e-th wtt reduction with partial recursive use function δ_e. We will meet R_e by attempting to code K into $B^{(e+1)}$, the $(e+1)^{\text{st}}$ column of B. Define $h(x)$ to be the least $y_1 > x$ such that for all $j \leq x$ and all strings σ of length x, there is a $y < y_1$ with $K_{y_1} \restriction y = K \restriction y$ and $\Delta_j(A, \langle j+1, y\rangle) \neq Q^\sigma(y)$, where

(i) $Q^\sigma(z) = \sigma(z)$ if $z < x$,

(ii) $Q^\sigma(z) = K(z)$ if $x \leq z < y_1$, (K is the standard 1–complete set), and

(iii) $Q^\sigma(z) = 0$ otherwise.

Note that as $A <_{wtt} \emptyset'$, $h(x)$ is defined (as Δ_j is a *wtt* reduction) and recursive in $\emptyset'$. The construction is again simple. We put $\langle 0, x\rangle$ into B iff $x \in A$ so that $A \leq_1 B$. Now suppose g is not dominated by h and is approximated by $g(x,s)$ as above. At stage s, for each $x \leq s$ and $j \leq x$ we set $B_s(\langle j+1, x\rangle) = K_{g(x,s)}(x)$. Note that this ensures that $B \leq_T A$, since A can compute $g(x)$ and then, to compute if $\langle j+1, x\rangle \in B$, we need only compute $K_{g(x)}(x)$.

To see that the construction succeeds for R_e, let $q > e$ be such that $g(q)$ exceeds $h(q)$. Compute a stage $s = s(q)$ where $g(q,t) = g(q)$ for all $t \geq s$, and with $g(q,s) = s-1$. Let σ equal $B^{(e+1)} \restriction q$. Now as $g(q) > h(q)$, there is a number y with $q \leq y < g(q)$ such that $\Delta_e(A; \langle e+1, y\rangle) \neq Q^\sigma(y)$. By construction, for all m with $q \leq m < g(q)$, $B_s(\langle j+1, m\rangle) = K_n(m)$ for some $n \geq g(q)$. As $K_{g(q)}(y) = K(y)$, we see that $B(\langle j+1, y\rangle) = Q^\sigma(y)$ as required. □

For the above result, the index set argument is perhaps simpler, but the construction used in our proof of (2.2) should be rather instructive.

We now give a new proof of Shoenfield's theorem. For this result, we feel our proof is rather more perspicacious than the original.

(2.3) Theorem. *(Shoenfield [1976]) Let A be r. e. and nonlow$_2$. Then there is an r. e. $D \equiv_T A$ with no hyperhypersimple superset.*

Proof: Let D be the standard deficiency set of A for the enumeration $f : D = \{s \mid \exists t > s(f(t) < f(s))\}$. Of course $D \equiv_T A$ (see Soare [1987, p.81]). Let D_t be the natural enumeration of D: At stage t we enumerate s if $f(t) < f(s)$ and s is not yet in D to get D_{t+1}. For our purposes, the crucial property of this enumeration is that if s is enumerated at t then all elements of the interval (s, t) not yet in D also enter D at t. (If $r \in (s, t)$ th

en $f(r) > f(s)$ as $s \notin D_{r+1}$ and so $r \in D_{t+1}$.) We call this property of an enumeration the *dump property*.

Now suppose that W is an r. e. coinfinite superset of D. We build a weak array $\{Q_e : e \in \omega\}$ that meets the following requirements.

$$R_e : |Q_e - W| \geq 1.$$

Let $h(x) = (\mu s)(s > x \;\&\; |[x, s] \cap \overline{W}| \geq 4x + 3)$.

Then $h \leq_T \emptyset'$ as $|\overline{W}| = \infty$. Take a $g \leq_T D$ not dominated by h with our conventional approximation $g(x, s)$. We also assume that $W_s \supseteq D_s$.

Construction, stage $s + 1$**:** Let $\gamma_0^s = 0$ and $\gamma_{i+1}^s = g(\gamma_i^s, s)$ for $i \leq s$. For each $j \leq s$ add elements to $Q_{i,s}$ (to get $Q_{i,s+1}$) for each $i \leq j$ in turn as follows: If $Q_{i,s} \subseteq W_s$ and there is an $x < s$ in $(\gamma_j^s; \gamma_{j+1}^s) \cap \overline{W_s}$ but not in $Q_{k,s+1}$ for $k < i$ or $Q_{m,s}$ for $m \neq i$, put the least such x into $Q_{i,s}$.

Verification Clearly the Q_i form a weak array. Let $\gamma_i = \lim_s \gamma_i^s$. The idea of the construction is, first, that infinitely often the intervals (γ_i, γ_{i+1}) contain many elements of $\overline{W}$ by our choice of g and its monotonicity. The second point is that the dump property will wipe out the effect of inappropriate enumerations into Q_i.

In particular, to see that R_e is met choose $x > \gamma_{e+2}$ with $g(x) > h(x)$ and let j be such that $\gamma_{j-1} < x \leq \gamma_j$. (Necessarily, $e < j \leq x$.) The monotonicity of g and the definition of the γ_i now guarantee that $\gamma_{j-1} < x < h(x) < \gamma_{j+1}$. By the definition of $h(x)$ then either $|(\gamma_{j-1}, \gamma_j) \cap \overline{W}| \geq 2x + 1$ or $|(\gamma_j, \gamma_{j+1}) \cap \overline{W}| \geq 2x + 1$. The argument is symmetric and we assume $|(\gamma_{j-1}, \gamma_j) \cap \overline{W}| \geq 2x + 1$. Let $s_{j-1}(s_j)$ be the stage at which $\gamma_{j-1}(\gamma_j)$ reaches its final value. By our conventions, $\gamma_{j-1} = s_{j-1}$ and $\gamma_j = s_j$. By construction, no numbers $> \gamma_{j-1} = s_{j-1}$ have been enumerated in any Q_i before stage s_{j-1}. By the dump property of our enumeration of D and our convention that $\gamma_j^s = g(\gamma_{j-1}, s)$, γ_j^s changes (at $s > s_{j-1}$) only when $D \upharpoonright \gamma_j^s$ changes. Thus we see that $(\gamma_j^{s_{j-1}}, \gamma_j) \subseteq D$ and so $|(\gamma_{j-1}, \gamma_j^{s_{j-1}}) \cap \overline{W}| \geq 2x + 1 \geq 2j + 1$. By our construction, the numbers in $(\gamma_{j-1}, \gamma_j^{s_j - 1})$ can be put into Q_i only for $i < j$ and only when $Q_{i,s} \subseteq W_s$. Thus each such Q_i, including Q_e, must eventually get an element of $(\gamma_{j-1}, \gamma_{j-1}^{s_j - 1}) \cap \overline{W}$ if not some other element of $\overline{W}$. □

An open question related to Theorem 2.3 is whether every nonlow$_2$ r. e. degree contains an r. e. set with no r–maximal superset (Soare [1987], p.233). Although we cannot prove this, we have found an easy proof that sets without r–maximal supersets exist.

(2.4) Theorem. *(Lachlan [1968]) There is a coinfinite r. e. set A with no r–maximal superset.*

Proof: The proof is similar in structure to Martin's proof of the existence of a coinfinite r. e. set with no maximal superset.

We divide ω into boxes as follows. W_0–boxes, $B_{0,i}$, consist of all pairs in order starting from one (i.e. $\{1,2\},\{3,4\},\dots$), so that $B_{0,i}=\{2i+1,2i+2\}$. W_{n+1}-boxes are defined inductively by $B_{n+1,i}=B_{n,2i+1}\cup B_{n,2i+2}$. Thus for each n and m, $B_{n,m}$ has 2^{n+1} elements. $B_{2,0}$, for example, is $\{7,8,...,13,14\}$. The construction is as follows: For all W_e boxes $B_{e,i}$, when all but one element is in $W_{e,s}$, put the other element into $A_{s+1}-A_s$.

Note that this means that at most $2^{n+1}-1$ elements of $B_{n,0}$ enter A so that $|\overline{A}|=\infty$. Finally, suppose $A\subseteq W_e$ and W_e is r–maximal. By our action, for each W_e–box B we have that $|B-W_e|\neq 1$. If, for almost all W_e–boxes we have $|B-W_e|=0$ then $W_e=^*\omega$. Hence, there are infinitely many W_e–boxes $B_{e,k}$ with $2\leq|B_{e,k}-W_e|\leq 2^{e+1}$. Pick the largest j with $2\leq j\leq 2^{e+1}$ such that there are infinitely many $B_{e,k}$ with $|B_{e,k}-W_e|=j$. Without loss of generality $|B_{e,k}-W_e|\leq j$ for all k.

We define sets Q and R as follows. For each W_e-box $B_{e,k}$, at the stage where $|B_{e,k}-W_{e,s}|=j$ put half of the j elements into Q and the rest of B_k into R. So $Q\cup R=\omega$ and $Q\cap R=\emptyset$. By construction, $|R-W_e|=|Q-W_e|=\infty$ so W_e is not r–maximal. $\square$

Remark. We note that that this construction is compatible with Martin's (e.g. Soare [1987, Ch. X Exercise 5.5]). We can even ensure that A has no hh–simple superset.

We finish this section with some nice applications of our technique to tt– and wtt–degrees. We begin by solving an old question on the complexity of minimal tt–degrees. (See e.g. Odifreddi [1981, 1989]).

We say a Turing degree $\mathbf{a}$ of some type Δ (e.g. r. e. or ω–r. e.) is Δ $(w)tt$–bottomed if there is an $A\in\Delta$ of degree $\mathbf{a}$ such that $A\leq_{(w)tt}B$ for every $B\in\mathbf{a}$ which is also in Δ.

(2.5) Theorem. *(i) If A is $\overline{low_2}$ and r. e. then there is an r. e. $B\equiv_T A$ such that $B\leq_{tt}A$ but $A\not\leq_{wtt}B$.*

(ii) In particular the following hold.

(a) No $\overline{low_2}$ r. e. degree is r. e. tt– or wtt–bottomed.

(b) No $\overline{low_2}$ r. e. degree contains a minimal r. e. tt–degree.

Proof: Call a string τ x–good if τ has at most x zeros. Let A be a given $\overline{low_2}$ r. e. set. Let Δ_e denote the e–th partial wtt reduction with use δ_e.

$$\begin{aligned}&\text{Let } h(x)=\max_{j\leq x}\min\{s:\forall\sigma(|\sigma|=s\text{ and }\sigma\text{ is }x\text{-good implies}\\&\exists y(\Delta_j(\sigma;y)\neq A(y))\text{ through a disagreement witnessed at stage }s,\\&\text{that is, }\Delta_{j,s}(\sigma;y)\downarrow\neq A_s(y)=A(y),\\&\text{or }K\text{ discovers that }\Delta_j(B;y)\text{ is not total for any }B\supseteq\sigma\text{ by stage }s\}.\end{aligned}$$

First let us explain how K discovers that $\Delta_j(B;y)$ is not total for any $B \supseteq \sigma$. For each y in turn, K first checks to see if the partial recursive use function δ_j associated with Δ_j is defined at y. If not $\Delta_j(B;y)\uparrow$ for every B. If $\delta_j(y)\downarrow < |\sigma|$, then K asks if $\Delta_j(\sigma;y)\downarrow$. If not $\Delta_j(B;y)\uparrow$ for every $B \supseteq \sigma$. If $\delta_j(y)\downarrow \geq |\sigma|$, no determination can be made about σ. Moreover, if, for any set C, $\Delta_j(C;y)$ is not total, this search procedure applied to all σ "simultaneously" eventually determines for some $\sigma \subseteq C$ that $\Delta_j(B;y)$ is not total for any $B \supseteq \sigma$.

Now consider the tree of x–good strings σ such that $\Delta_j(\sigma;y)$, where defined, gives an initial segment of A and K cannot determine that $\Delta_j(B;y)$ is not total for every $B \supseteq \sigma$. If this tree were infinite then, by König's lemma, it would have an infinite path defining a set B. By the definition of the tree, B would have at most x many zeros and $\Delta_j(B;y) = A(y)$ for every y. As this implies that A is recursive, the tree is finite and so $h(x)$ is well defined. As $h(x) \leq_T K$, there is a $g \leq_T A$ not dominated by $h(x)$.

We build B via a standard movable marker construction where at stage s we move the x–th marker to a number bigger than $g(x,s)+1+s$ when $A \restriction g(x,0)$ changes at s, dumping the old position and all intervening elements into B. Note that (vi) of the conventions for $g(x,s)$ implies that we always move the x-th marker at least once. Thus it is clear from the construction that the principal function $b(x)$ of $\overline{B}$ dominates $g(x)$ and hence $b(x)$ is not dominated by h. Now A and B have the same T–degree: to compute $A(x)$ from B run the construction till B is correct on the first x many members of its complement. $A(x)$ cannot change again by the definition of B. Moreover, $B \leq_{tt} A$. To see if $x \in B$, look at stage $x+1$ of the construction. If x is not yet in B then x is the position of some marker y. We claim that $x \in B$ iff $A \restriction g(y,x{+}1) \neq A_{x+1} \restriction g(y,x{+}1)$. If $A \restriction g(y,x{+}1) = A_{x+1} \restriction g(y,x{+}1)$, then by construction and convention (ii), (iv) none of the markers $\leq y$ ever move again and so $x \notin B$. On the other hand, if $A \restriction g(y,x+1) \neq A_{x+1} \restriction g(y,x+1)$ then, by construction, some marker $z \leq y$ is moved at a stage $s > x+1$. It is moved to a number $> s > x$ and so x is put into B.

Finally, we claim that $A \not\leq_{wtt} B$. Consider any wtt reduction Δ_j with $\Delta_j(B;y)$ total. We now choose an $x > j$ with $b(x) > h(x)$. As $B \restriction b(x)$ is an x–good string (remember $b(x)$ is the x^{th} element of $\overline{B}$ so $|\overline{B} \restriction b(x)| = x-1$) of length $\geq h(x)$ and $\Delta_j(B;y)$ is total, $\Delta_j(B \restriction b(x);y)\downarrow \neq A(y)$ for some y by the definition of h. □

We note that Andre Nies had previously found the following rather pretty index set proof of (2.5) (ii)(a).

Alternative proof of (2.5) (ii)(a) (A. Nies)

Let A be a given r. e. $\overline{low}_2$ set. Suppose A is the (w)tt–bottom of the T–degree of A and $S \in \Sigma_3^A$. Then by Yates' index theorem (see Soare [1987], XII.1.5) there is a recursive function g such that for every x, $W_{g(x)} \leq_T A$ and $x \in S$ iff $W_{g(x)} \equiv_T A$.

Now, as A is the tt–bottom of the Turing degree of A, $W_{g(x)} \equiv_T A$ is equivalent to $A \leq_{(w)tt} W_{g(x)}$ since $W_{g(x)} \leq_T A$. As the reductions are Σ_3, this means $S \in \Sigma_3$ and hence A is low$_2$. □

Now (2.5) has several extensions. First we observe that it works for sets A which are ω–r. e. in the sense of Ershov's difference hierarchy. These are the sets A with approximations $A(x) = \lim_s A_s(x)$ for which there exists a recursive function h

such that for all x, $|\{s : A_{s+1}(x) \neq A_s(x)\}| \leq h(x)$. They are also the sets A which are tt–reducible to K. The reader should consult e.g. Epstein, Haas, and Kramer [1981] for basic facts about these sets. One simply emulates our proof of (2.5) using such an approximation for A and a standard Δ_2^0 marker construction. That is, we move the marker for x to a number bigger than $g(x,s)+1+s$ when $A_s \restriction g(x,s) \neq A_t \restriction g(x,s)$ for any $t < s$. If $A_s \restriction g(x,s)$ returns to a previous configuration (say to $A_{s_1} \restriction g(x,s)$ with $s_1 < s$), we must return the marker for x back to the place it had at stage s_1. With this modification, we see

(2.6) Corollary.

(i) If A is any $\overline{low_2}$ ω–r. e. set then there exists an ω–r. e. $B \equiv_T A$ with $B \leq_{tt} A$ yet $B \not\leq_{wtt} A$.

(ii) In particular no $\overline{low_2}$ ω–r. e. degree is ω–r. e. $tt-$ or wtt–bottomed nor does it contain a minimal ω–r. e. tt–degree.

We conclude with one further extension of (2.5).

(2.7) Theorem.

(i) The $\overline{low_2}$ r. e. tt–degrees are dense.

(ii) So too are the $\overline{low_2}$ ω–r. e. tt–degrees.

Remark. Other variations are again possible. For instance one can additionally avoid cones.

Proof: We only prove (i). The proof of (ii) is translation of that of (i) to the realm of Δ_2 approximations similar to, but more complicated than, that given for Theorem 2.5 above. Details can be found in Nies and Shore [ta].

Suppose we are given $\overline{low_2}$ r. e. sets $B <_{tt} A$. We will construct C r. e. with $B <_{tt} B \oplus C <_{tt} A$. For each potential tt reduction Δ_e, we aim to satisfy the requirements

$$R_{2e} : \Delta_e(B) \neq C$$

$$R_{2e+1} : \Delta_e(B \oplus C) \neq A.$$

Following ideas from the proof of the density theorem of Ladner [1975] in the polynomial time degrees, we meet R_{2e} by making C locally copy A and we meet R_{2e+1} by making C locally look like ω. To meet R_{2e} we code A in $C^{(q)}$ for some q. We perform the following process.

Define $l(2e,q,s) = \max\{x : (\forall y < x)(\Delta_{e,s}(B_s; \langle q,y\rangle) = C_s(\langle q,y\rangle))\}$. Initially we set $C_x(\langle q,x\rangle) = 1$ for each stage x until a stage s_0 occurs where $l(2e,q,s_0) > 0$. At the first such stage s_0 we make $\langle q,s_0\rangle$ the coding marker for 0 in $C^{(q)}$, i.e., $\langle q,s_0\rangle \in C$ iff $0 \in A$. More generally, if we have already defined coding markers $\langle q,s_0\rangle,\ldots,\langle q,s_{x-1}\rangle$ for $0,\ldots,x-1$ in $C^{(q)}$ we set $C(\langle q,y\rangle) = 1$ for stages $y > s_{x-1}$ until a stage s_x occurs with $l(2e,q,s_x) > s_{x-1}$. At the least such stage s_x, we make $\langle q,s_x\rangle$ the coding marker for x in $C^{(q)}$, i.e., $\langle q,s_x\rangle \in C$ iff $x \in A$. Note that this process makes $C^{(q)} \leq_{tt} A$ uniformly in q: To compute $C(\langle q,s\rangle)$ run the construction for $\langle q,s\rangle+1$ many stages. If $\langle q,s\rangle$ has not been declared a coding marker, $\langle q,s\rangle \in C$. If it has been made the coding marker for x in $C^{(q)}$ then $\langle q,s\rangle \in C$ iff $x \in A$.

We claim that $\Delta_e(B) \neq C$ and, in particular, there is an x such that $\Delta_e(B;\langle q,x\rangle) \neq C(\langle q,x\rangle)$. If not, $l(2e,q,s) \to \infty$ and so, for each x, there is a marker $\langle q,s_x\rangle$ (which we can find recursively) such that $x \in A$ iff $\langle q,s_x\rangle \in C$, i.e., $A \leq_{tt} C$. As this would imply that $A \leq_{tt} B$, we have the desired contradiction. Next, note that $\Delta_e(B)^{(q)} \neq C^{(q)}$ implies that $\overline{\lim}\ l(2e,q,s) < \infty$ and so, by construction, $C^{(q)}$ is cofinite. Finally, K can compute an index for $C^{(q)}$ as a cofinite set. (To make the index of the requirement visible, we would label this set $C^{(q)}_{2e}$.)

Suppose R_0 is met by applying the above procedure with $q = 0$ $(= q(0,s))$. To meet R_1, the basic idea is to keep setting $C^{(d)} = \omega^{(d)}$ for as many $d > 0$ as necessary to guarantee that $\Delta_0(B \oplus C) \neq A$. As this procedure threatens to make C recursive ($C^{(0)}$ is co finite), $A \not\leq_{tt} B$ and Δ_0 is a possible tt–reduction, the procedure must eventually succeed. Moreover, K can find a number d such that setting $C^{(0)} = C^{(0)}_0$ and $C^{(i)} = \omega^{(i)}$ for $0 < i < d$ guarantees that $\Delta_0(B \oplus C) \neq A$. This d is the value of $h(0)$ for the function $h \leq_T K$ for which we will apply the nonlow$_2$–ness of A as in the previous arguments.

We expect to define a $g \leq_T A$ not dominated by h and a recursive approximation $g(x,s)$ as above. The plan is to make $C^{(y)} = \omega^{(y)}$ for all $1 \leq y \leq g(0)$ by putting all elements of $\omega^{(y)}$, $1 \leq y < s$ into C when $A \restriction g(0,s)$ changes. Indeed, we set $C^{(y)} = \omega^{(y)}$ for every y for which we have made any commitments for $C^{(y)}$. This guarantees that $C^{(y)} = \omega^{(y)}$ for all y, $1 \leq y \leq g(0)$ by our conventions on $g(x,s)$. Thus if $g(0) > h(0)$, we meet R_1.

In this environment we meet R_2 by following our strategy with the appropriate q. We let $q(2,s)$ be the first column not yet filled in by our actions for R_1. We follow our strategy for the q at stage s. If t is the last stage at which $A \restriction g(0,s)$ changes then $q(2,t) = \lim_s q(2,s) = q(2)$. We thus satisfy R_2 by our construction of $C^{(q(2))}$.

Now we again turn to the odd requirements. First, note that we may not yet have met R_1 (if $g(0) < h(0)$). We must define $h(1)$ so large that filling in $C^{(y)}$ for $q(2) < y \leq h(1)$ would satisfy both R_1 and R_3. Note that we cannot, however, use the recursion theorem as h must be given directly recursively in $\emptyset'$. Our first task is to see how far we would have needed to make $C^{(y)} = \omega^{(y)}$ to meet both R_1 and R_3 (given that $C^{(0)} = C^{(0)}_0$). Suppose d_0 suffices. If $q(2) \geq d_0$, our action for R_1 would actually meet both R_1 and R_3. We next consider the possibility that some $y < d_0$ is $q(2)$, the column on which we meet R_2. To do this, we compute (indices for) $C^{(y)}_2$ for $1 \leq y \leq d_0$ and then, for each y, we find $d_{1,y}$ such that meeting R_0 at $C^{(0)}$ and R_2 at $C^{(y)}$ and all other columns being ω, making $C^{(0)} = C^{(0)}_0$, $C^{(z)} = \omega^{(z)}$ for $1 \leq z < y$, $C^{(y)} = C^{(y)}_2$ and $C^{(z)} = \omega^{(z)}$ for $y < z \leq d_{1,y}$ meets both R_1 and R_3. We let $h(1) = d_1 = \max\{d_{1,y} \mid y \leq d_0\}$.

Our action for R_3 is much like that for R_1. When $A \restriction g(1,s)$ changes, we make $C^{(y)} = \omega^{(y)}$ for all $y > q(2,s)$ for which we have ever made any commitments. Once again we argue that if $g(1) > h(1)$ we satisfy both R_1 and R_3: Either $q(2) \geq d_0$, in which case we won already, or $q(2) \leq d_0$. In this case, we also know by our definition of h that $C^{(0)}_0 \cup C^{(q(2))}_2 \cup \omega^{(k)}$ for $k \leq h(1)$, $k \neq q(0)$, $q(2)$ forces a win for both R_1 and R_3. Moreover, if $g(1) > h(1)$ but $g(0) \leq q(2) < d_0$, then it is clear that our construction produces this result for $C^{(k)}$, $k \leq d_0$.

Of course, we satisfy R_4 at the first column $q(4,s)$ not used by our actions for R_3. The general plan for defining $h(e)$ and satisfying R_{2e+1} (indeed R_{2i+1} for $i \leq e$) is as for $e = 1$. We first compute d_0 large enough to meet all of $R_1, \dots, R_{2e+1}$. We then compute d_1 large enough to win all of them on the assumption that exactly one $y < d_0$ is used for coding (i.e., is $C_0^{(y)}$). We then compute d_2 on the assumption that exactly two columns below d_1 are used as $C_0^{(y_0)}$ and $C_2^{(y_2)}$. In this way, we compute $d_0, d_1, \dots, d_e$. The required $d_e = h(e)$ is $\max\{d_i \mid i < e\}$.

The crucial observation is that for any $g(e)$ exceeding $h(e)$, no matter what the previous pattern, we will be able to win all of $\Delta_0, \dots, \Delta_e$. Thus we meet all the requirements.

To see that $C \leq_{tt} A$, consider computing $C(\langle q, x\rangle)$. Run the construction until a stage s at which $C^{(q)}$ is set equal to $\omega^{(q)}$ or $q = q(2e, s)$ and $\langle q, x\rangle$ is either already put into C or is assigned as a coding marker for some $y \in A$. (It should be clear from the construction that one of these events eventually occurs.) Suppose n is the largest number such that $g(n, s) \leq q$. If $A \restriction g(n, s)$ ever changes, i.e., $A_s \restriction g(n, s) \neq A \restriction g(n, s)$ then all of $\omega^{(q)}$ is put into C. If not, then $q = q(2e)$ and $\langle q, x\rangle \in C$ iff $y \in A$. This procedure is the required truth–table reduction. $\square$

3. *Low*$_2$ results.

In this section we prove some results on low_2 r. e. degrees. One natural question arising from the results of §2 is whether the low$_2$ classes arising there are strictly low. For instance we know that all incomplete *wtt*–topped r. e. degrees are low_2. Are they in fact all low? Here we know the answer is no, since Downey and Jockusch [1987] showed that there exist incomplete 1–topped r. e. degrees, and all such degrees are low$_2$–low$_1$. Similarly, Lachlan [1968] showed that all coinfinite low$_2$ r. e. sets have maximal supersets so that (2.3) is sharp in terms of the high/low classification scheme. Downey [1993] constructs a low$_2$–low$_1$ array recursive degree and therefore (2.1) is tight too. The last of the results of §2 dealt with *tt*– bottomed degrees. Again, the result is sharp.

(3.1) Theorem. *There exist low$_2$–low$_1$ wtt– bottomed T–degrees containing r. e. sets of minimal tt–degree.*

Proof sketch: Originally, we had a direct proof of this result, but subsequently discovered that one can modify known results to obtain the desired one. The construction of Downey–Slaman [1989] of a completely mitotic promptly simple contiguous degree can easily be combined with the construction of Downey–Jockusch [1987], Theorem (4.3) and Theorem (4.17), of a strongly contiguous strongly *tt*–bottomed r. e. degree. Let A be the set so constructed. Then by Downey–Slaman [1989, Theorem 2.1], A is low$_2$–low$_1$, since this theorem says no low set of promptly simple degree is completely mitotic. The result then follows by the following observation by Nies (personal communication).

(3.2) Lemma. *(Nies) Suppose A is the strong tt–bottom of an r. e. T–degree. Then A has minimal r. e. tt–degree.*

Proof: (Nies) Let D be the deficiency set of A. Then $D \leq_{tt} A$ and $A \leq_T D$, hence $D \equiv_{tt} A$. As D is simple and semirecursive, $X \leq_{tt} D$ and $X \not\equiv_T \emptyset$ implies $D \leq_T X$ (Degtev [1978], see Odifreddi [ta, Proposition X.7.12]). Thus $A \equiv_T X$ and $A \equiv_{tt} X$. So we see that A has minimal *tt*–degree. $\square$

We now give some new and rather surprising characterisations of an r. e. set being low$_2$ in terms of the structure of m and *tt*–degrees. The first result shows that every low$_2$ r. e. set has a minimal cover in the r. e. *tt*–degrees. Together with Theorem 2.7, this theorem shows that the low$_2$ r. e. *tt*–degrees are definable in the r. e. *tt*–degrees. We view this result as the highlight of the paper. The second result improves a claim of Downey and Jockusch [1987, Theorem 3.5]. They claimed that if $\mathbf{c}$ is r. e. and low$_1$ then there is a 1–topped incomplete r. e. $\mathbf{a} \geq \mathbf{c}$. Although the proof there is incorrect, we in fact show that the result is correct for every low$_2$ r. e. $\mathbf{c}$. The proof employs a more complicated version of the techniques used for the first result. As all 1–topped r. e. sets are low$_2$, it proves the definability of the low$_2$ degrees in a language with two orderings, one for the Turing degrees and the other for either many–one or one–one degrees.

(3.3) Theorem. *(i) If C is r. e. and low$_2$, then there exists an r. e. A such that the tt–degree of A is a minimal cover of the tt–degree of C.*

(ii) Consequently, the jump class low$_2$ is definable in the r. e. tt–degrees: $\mathbf{a}$ *is low$_2$ iff* $\mathbf{a}$ *has a minimal cover.*

(iii) The same result holds for the tt–degrees below $\mathbf{0}'$ *and the wtt–degrees below* $\mathbf{0}'$.

Proof: (i) Let C be r. e. and given. We need to build A to meet the following requirements.

$$P_e : \Phi_e(C) \neq A$$

R_e : If Φ_e is a total *tt*–reduction, then $\Phi_e(A \oplus C) \leq_{tt} C$ or $A \oplus C \leq_{tt} \Phi_e(A \oplus C) \oplus C$.

Here Φ_e is the e–th possible *tt*–reduction and has use φ_e. In this construction we modify the ideas of Kobzev [1979], Degtev [1973] and Fejer–Shore [1989]. At each stage s, let $\{a_{0,s}, a_{1,s}, \dots\}$ list $\overline{A}_s$ in order. We begin by discussing the method whereby we construct a minimal r. e. *tt*–degree when $C = \emptyset$.

Let $\ell(e,s) = \max\{z : (\forall y < z)(\Phi_{e,s}(A_s \oplus C_s; y)\downarrow)\}$. The construction can either be performed by intersecting trees as in Fejer–Shore [1989] or can be thought of as an "η–maximal set" type construction where we build equivalence classes as in Odifreddi [1989, pp.302–318]. We choose the latter. Our construction will use the *dump* method. That is, if $A_{s+1} \neq A_s$ then, for some i, $A_{s+1} = A_s^i$ where $A_s^i = A_s \cup \{a_{j,s} : i \leq j \leq s\}$.

For a fixed e, the strategy is as follows. During the construction we dynamically build (finite) equivalence classes which we refer to as *boxes*. Initially $B_{j,0} = \{a_{j,0}\}$ and we always let $b_{j,s} = a_{d(j),s}$ denote the first element of $B_{j,s}$. At each stage s, $B_{j,s} \cap A_s = \emptyset$. The primary rule of the construction is that at stage s, if any of $B_{j,s}$ enters A_{s+1} then all of $B_{j,s}$ enters. Finally, a box will have a *state*. In the simple construction — that is with $C = \emptyset$ — box B_e will have an e*–state*, a string of length e. This state will be used to encode the $j \leq e$ for which we have seen j–splittings. It will be defined formally below but the crucial fact will be the usual one for e–state arguments. For each e the e–state of box $B_{e,s}$ is eventually constant and once the i–state of $B_{i,s'}$ has reached its final value for $i \leq e$, the box $B_{e,s}$ itself is also constant. As we use its first element, $b_{e,s}$, as our witness for satisfying the Friedberg requirement P_e in the usual way, this fact will assure the success of P_e. More precisely, we wait for a stage s such that $\Phi_e(C_s(=\emptyset); b_{e,s}) = A_s(b_{e,s}) = 0$. At such a stage, we would put $b_{e,s}$ into A, winning the requirement forever. As $b_{e,s}$ is eventually constant, the procedure satisfies P_e. Of course, in the real construction,

C will later be able to change necessitating further attacks on P_e. The reader should, however, note that the important dynamic of the construction is achieving a stable follower $b_{e,s}$ from a box $B_{e,s}$ in the "correct" e–state, i.e., the one that correctly codes whether or not we get infinitely many j–splittings for each $j \leq e$. In the full construction, this follower will need to be chosen so that our actions will be sufficiently C–correct to win by using it.

We now define the e–state of a box B_i for $i \geq e$ and describe how to increase it. The e–state $\sigma(e,i,s)$ of $B_{i,s}$ at s has value 1 at $j \leq e$ if there seems to be a *j–splitting for $B_{i,s}$ at z* for some $z < s$, i. e., $\exists z < s \Phi_j(A_s \oplus C; z)\downarrow \neq \Phi_j(A_s^{d(i)} \oplus C; z)\downarrow$ at stage s and the use $\varphi_j(z)$ is less than $b_{i+1,s}$ (and so if $B_{i,t}$ does not change then neither does $A_t \restriction \varphi_j(z)$). Otherwise, it has value 0 at j. Consider the simple case $C = \emptyset$. The requirement R_e attempts to maximize the e–states of boxes (in the usual lexicographic ordering). It finds the first j and k greater than e such that $j < k$ but $\sigma(e,j,s) < \sigma(e,k,s)$ and raises the e–state of B_j by amalgamating boxes $B_{j-1}, \dots, B_{k-1}$:

$$\begin{aligned} B_{i,s+1} &= B_{i,s} \quad i < j-1 \\ B_{j-1,s+1} &= \bigcup_{j-1 \leq i < k} B_{i,s} \\ B_{j+n,s+1} &= B_{k+n,s} \quad n \geq 0 \end{aligned}$$

This action keeps the e–state of $B_{i,s+1}$ the same as that of $B_{i,s}$ for $i < j$ but makes that of $B_{j,s+1}$ be $\sigma(e,k,s) > \sigma(e,j,s)$. (Of course, the e–state of a box may increase "on its own" as time goes by and more splittings are discovered. Such increases call for no additional action on our part.) The usual e–state arguments show that all boxes are finite in the limit and that almost all of them are in the same e–state. We call this the *well–resided* e–state. The crucial point here is that, once we have acted for each P_i, $i \leq e$, for which we ever act (say by stage s_0) we change $B_{i,s}$ (by making it larger) for $i \leq e$ only to increase its i–state and we never put $B_{i,s}$ into A for $i \leq e$. Suppose $B_{i,s}$ and its i–state are constant for $s \geq s_1 \geq s_0$ and consider B_e. If at $s \geq s_1$ we have $\sigma(e,j,s) = 1$ for some $j \leq e$, we have $\Phi_j(A_s \oplus C; z) \downarrow \neq \Phi_j(A_s^{d(e)} \otimes C; z) \downarrow$ and $\varphi_j(z) < b_{e+1,s}$. As we can only increase $B_{e,t}$ for $t \geq s$ and no $B_{i,t}$ ever enters A for $i \leq e$, $A_t \restriction \varphi_j(z) = A_s \restriction \varphi_j(z)$ for $t > s$. Of course $b_{e,t}$ is also constant and so $A_t^{d(e)} \restriction \varphi_j(z) = A_s^{d(e)} \restriction \varphi_j(z)$ for every $t \geq s$ as well. Thus the j–splitting for B_e at z can never disappear after s and so the e–state of $B_{e,s}$ can never decrease after s_1. As usual, this means that $B_{e,s}$ and its e–state are eventually constant for each e. These facts suffice to establish the theorem when $C = \emptyset$. To see this, we prove that the requirements R_e are satisfied. Let η be the well–resided $(e+1)$–state. If $\eta(e) = 1$ we will prove that $A \leq_{tt} \Phi_e(A)$. If $\eta(e) = 0$ we will prove that $\phi_e(A) \leq_{tt} C = \emptyset$. In either case we describe the required reductions for x larger than all elements of the n many final boxes which are not in the well–resided $(e+1)$–state η. Suppose first that $\eta(e) = 1$. Find a stage s by which $B_{i,s}$ have settled down for $i < n$ and x is either in A or in a box $B_{j,s}$ with state η. Suppose $x \in B_{j,s}$. Now x will later enter A if and only if $B_{k,s}$ is put into A for some k, $n \leq k \leq j$. As $\eta(e) = 1$, there is for each k, $n \leq k \leq j$, a number $z(k)$ such that $\Phi_{e,s}(A_s; z(k)) \neq \Phi_{e,s}(A_s^{d(k)}; z(k))$. We can thus check for each such k in turn if $B_{k,s}$ enters A by answering the appropriate question about $\Phi_e(A)$. Clearly this provides a tt–reduction for A from $\Phi_e(A)$.

If $\eta(e) = 0$ and Φ_e is a tt–reduction, we claim $\Phi_e(A)$ is recursive. To compute $\Phi_e(A;x)$ let n be as above and let s be a stage by which all B_i, $i \le n$, have settled down, $\Phi_{e,s}(A_s;x)\downarrow$ and all $B_{j,s}$ are in state η for $j \ge n$ and $b_{j,s} < \varphi_{e,s}(x)$. We claim $\Phi_{e,s}(A_s;x) = \Phi_{e,t}(A_t;x)$ for all $t \ge s$ and so $\Phi_{e,s}(A;x) = \Phi_e(A;x)$. If not, let $t+1$ be the first counterexample, i.e., for some $i \ge n$,

$$\Phi_{e,t+1}(A_{t+1};x) = \Phi_{e,t}(A_t^{d(i)};x) \ne \Phi_{e,s}(A_s;x) = \Phi_{e,t}(A_t;x).$$

Of course $a_{d(i),t} < \varphi_{e,t}(x)$. Thus we could raise the state of B_n at t to take value 1 at e, contrary to our choice of n and s.

(We note that this argument can be modified to make $A \le_m \Phi_e(A)$ if $\Phi_e(A)$ is r. e. by the methods of Downey [1989].)

We now turn to the case with C low$_2$. We employ a tree construction and assume familiarity with such constructions as presented in Soare [1987, XIV]. Familarity with earlier low$_2$ constructions as in Shore and Slaman [1990] would also be helpful. We will meet the requirements $P_e : \Phi_e(C) \ne A$ via a Sacks coding strategy. The version α of P_e with the guess τ at the well resided e–state will attempt to code K up to the length of agreement with A. Let $L(\alpha, s)$ be the length of agreement function for P_e. The node α takes control of the boxes $B_{i,s}$ with (the code for) $\alpha < i < L(\alpha, s)$ which are not controlled by nodes of higher priority and are in state τ. (So, in particular, no action can be taken by nodes of lower priority to improve their j–state for $j > e$.) If one such i later enters K we put $B_{i,s}$ into A. If Φ_e is a total truth–table reduction we must eventually produce a disagreement as $K \not\le_T C$. If not, $L(\alpha, s)$ is eventually constant anyway. Thus α's effect on nodes extending it is finitary. Note that if we move to the left of α, it loses its control over all boxes.

To be more precise, suppose by induction that α is on the leftmost path and that on the C–true α–stages (defined below) the set of boxes assigned to each node of higher priority than α is eventually constant and that, from some point on, no $\beta < \alpha$ ever acts to redefine any box. Moreover, assume also by induction that on the C–true α–stages almost all boxes eventually assigned to α are in state τ. If Φ_e is not a truth–table reduction, the length of agreement function is obviously eventually constant and so, therefore, is the set of boxes eventually controlled by α on the C–true α–stages. If Φ_e is a truth–table reduction, then any box $B_{i,s}$ controlled by α at a sufficiently large C–true α–stage s is controlled by α at every later C–true α–stage and it is eventually fixed by construction and our induction hypothesis. If $\Phi_e(C) = A$, α would then eventually control $B_{i,s}$ for every i larger than some i_0 at every C–true α–stage. Once so controlled, $B_{i,s}$ will enter A only if some $j \le i$ enters K. In this case, we could compute K from $A \oplus C$ and so from C for our desired contradiction. Thus $\Phi_e(C) \ne A$ and once again the length of agreement is eventually constant. In any case, the set of boxes controlled by α is eventually constant and, from some point on, α does not act to redefine any boxes (by putting numbers into A) so each box is eventually fixed as well.

We can thus consider 0 to be the only outcome of α and make $\alpha\widehat{\ }0$ accessible whenever α is. Of course $\alpha\widehat{\ }0$ works on the next R requirement.

Thus the crucial point, for $C \neq \emptyset$, is determining when a node η, such as $\alpha\hat{}0$, trying to maximize the state of boxes at e for the sake of R_e should act by amalgamating boxes. (It has an assumption τ as to the well-resided $(e-1)$–state.) The problem, of course, is that it may appear that $\sigma(e-1,j,s) = \sigma(e-1,k,s) = \tau$ and $\sigma(e,j,s) < \sigma(e,k,s)$ but the computation $\Phi_e(A_s \oplus C_s; z)\downarrow \neq \Phi_e(A_s^{d(k)} \oplus C_s; z)\downarrow$ may be C–incorrect. Acting on the basis of such computations could result in infinitary action without stabilizing the $(e+1)$–states. Our basic plan is to use low$_2$–ness in the fashion of Slaman and Shore [1990] to approximate the answers to the appropriate question of C'' at node η:

$$Q(\eta) : \forall n \exists^\infty s \; (s \text{ is a } C - \text{true } \eta - \text{stage and there is a box } B_m, m > n,$$
$$\text{in state } \tau\hat{}1 \text{ via computations using only } C \restriction c(\eta, s)?$$

($c(\eta, s)$ is defined below and τ is η's guess at the well–resided e–state.)

Notation: We approximate C as $C_s = \{c(t) \mid t < s\}$ where $c(s)$ is a 1–1 recursive enumeration of C. If η is accessible at s and was last accessible at r $(= 0$ if η has never been accessible before), then $c(\eta, s)$ is $\min(C_s - C_r)$, the least number enumerated C since stage r. We say that a stage s is a *C–true η–stage* if η is accessible at s and $C_s \restriction c(\eta, s) = C \restriction c(\eta, s)$. This procedure is a version of Lachlan's "hat trick" (Soare [1987], p.131) done on a tree.

In the usual way, we use the recursion theorem to uniformly get an index for the C'' question $Q(\eta)$ and so, by the low$_2$–ness of C, a Δ_3 index for the answer, i.e. we have an index a such that for every η, if the answer to our question is yes, then $\{a\}^{\emptyset''}(\eta) = 1$ and otherwise $\{a\}^{\emptyset''}(\eta) = 0$. The typical approximation procedure for Δ_3 functions then gives us a recursive g such that for every η the second coordinate (π_2) of $\liminf g(\eta, s)$ is $\{a\}^{\emptyset''}(\eta)$. [As $h(\eta) = \{a\}^{\emptyset''}(\eta)$ is a Δ_3 function, we may choose recursive predicates R_0 and R_1 such that, for $k = 0, 1$, $h(\eta) = k$ iff $\exists x \forall y \exists z R_k(x, y, z, a, \eta)$. The idea is to define $g(\eta, s)$ as the least $\langle x, k\rangle$ for which we have made progress at stage s towards seeing that $\forall y \exists z R_k(x, y, z, a, \eta)$. Formally we choose the least $\langle x, k\rangle$ such that either there is no $r < s$ for which $g(\eta, r) = \langle x, k\rangle$ or $u = \mu y(\neg(\exists z < s) R_k(x, y, z, a, \eta))$ is larger than the corresponding number at the last t for which $g(\eta, t) = \langle x, k\rangle$ (that is, such that if t is the largest r less than s such that $g(\eta, r) = \langle x, k\rangle$ and $v = \mu y(\neg(\exists z < t) R_k(x, y, z, a, \eta))$, then $u > v$). It is clear that $g(\eta, s)$ is always defined by the first alternative if not by the second. It is easy to see that our desired function can be taken to be $g(\eta, s)$.] It is this function $g(\eta, s)$ that we use to approximate the required answer to our C'' questions.

Next we describe the outcomes of η which is working on R_e. As a first approximation, the outcome at stage s of the node η should, in essence, be the value $\langle x, k\rangle$ of $g(\eta, s)$. Note first, that we must coordinate the approximations to g at successive nodes by the standard procedure that calculations at a node η work on the assumption that the only stages of the construction to be considered are those at which η is accessible. In terms of bookkeeping, one could try to simply consider $g(\eta, t)$ where, at s, η has become accessible for the t–th time. The left to right ordering on these outcomes is the usual lexicographic ordering on pairs. A further coordination procedure, however, is also required.

Suppose η (with guess τ at the well resided $e-1$–state) is accessible at stage s for the t^{th} time and was last accessible at r. If $g(\eta, t) = \langle y, k\rangle$, we assign a chip $\langle \eta, y, k, s\rangle$ to the corresponding outcome $\eta\widehat{\ }\langle y, k\rangle$ of η. These chips will be used (by assigning various boxes to them) to monitor the number of times and the priority with which the various successors of η are allowed to act either to preserve computations by keeping various boxes out of A or to code K by putting boxes into A. Let B_{j_0} be the last box controlled by a requirement of priority greater than η and $B_{j_1}, \dots, B_{j_n}$ be the boxes not controlled by any requirement of higher priority than η which are in state $\tau\widehat{\ }1$ (all at stage s). Assign all the boxes in each of the following intervals of boxes $(B_{j_0}, \dots, B_{j_1}), [B_{j_1}, \dots, B_{j_2}), \dots, [B_{j_{n-1}}, \dots, B_{j_n})$ to a chip $\langle \eta, x, 1, v\rangle$, $v \leq s$, in order of priority (lexicographic order on the chips). (Thus for example B_i is assigned to the first chip of the form $\langle \eta, x, 1, v\rangle$ which is assigned to $\eta\widehat{\ }\langle x, 1\rangle$ for $j_0 < i < j_1$; for $j_i \leq i < j_2$, B_i is assigned to the second such chip.) The outcomes $\langle \eta, x, 0, v\rangle$ do not expect any boxes in state $\tau\widehat{\ }1$ and so do not get any of the boxes in state $\tau\widehat{\ }1$ assigned to their chips. Instead, we simply assign the boxes $B_{j_n+1}, B_{j_n+2}, \dots$ in order to the chips assigned to $\eta\widehat{\ }\langle x, 0\rangle$. These boxes are now protected with the priority of the chip $\langle \eta, x, k, v\rangle$ to which they are assigned against any action by requirements to the right. This protection remains in effect as long as the relevant C–computations are correct and no node $\beta <_L \eta$ is accessible. It does not, however, interfere with the action of $\eta\widehat{\ }\langle x, 1\rangle$ or its successors. In particular, the boxes assigned to chips $\langle \eta, x, 1, v\rangle$ (actually, the ones resulting from the amalgamation and dumping action of $\eta\widehat{\ }\langle x, 1\rangle$ described below) are the ones available to nodes extending $\eta\widehat{\ }\langle x, 1\rangle$. Similarly, the ones assigned to chips $\langle \eta, x, 0, v\rangle$ can be passed on to nodes extending $\eta\widehat{\ }\langle x, 0\rangle$.

The determination of the accessible immediate successor of η depends primarily on the approximation procedure $g(\eta, t)$. It must, however, be coordinated with the C–true η–stages as well. Let r be the last stage that appears to be a C–true η–stage, i.e., the largest r such that $c(t) > c(\eta, r)$ for $r < t \leq s$. We declare $\eta\widehat{\ }\langle x, k\rangle$ to be accessible where $\langle x, k\rangle$ is the lexicographically least value of $g(\eta, u)$ with $r \leq u \leq s$. If $k = 1$ we amalgamate boxes $B_{j_i}, \dots, B_{j_{i+1}-1}$ for the sequences assigned to chips $\langle \eta, x, k, v\rangle$; thereby making the e state of B_{j_0+i} be $\tau\widehat{\ }1$. (Note that if there are no C–changes and we never move left of $\eta\widehat{\ }\langle x, 1\rangle$, then the new boxes remain in state $\tau\widehat{\ }1$. Each of them, as a sequence of length one, will automatically be assigned to its own chip $\langle \eta, x, 1, w\rangle$ at the next η stage.) If at $s' > s$ the computation witnessing the high e–state is seen to be C–incorrect, we dump boxes $B_{j_i+1}, \dots, B_{s'}$ into A. If $k = 0$, we take no action for η but protect the boxes assigned to its chips $\langle \eta, x, 0, v\rangle$ from any action by nodes to the right of $\eta\widehat{\ }\langle x, 0\rangle$ until some node to its left becomes accessible. (Again no actions by node $\alpha \supseteq \eta\widehat{\ }\langle x, 0\rangle$ are restricted.)

In any case, we now act for $\eta\widehat{\ }\langle x, k\rangle$ which is assigned to some P_j. Our action at $\eta\widehat{\ }\langle x, k\rangle$ for P_j is to code K (up to the appropriate length of agreement) on the sequence of boxes handed down to $\eta\widehat{\ }\langle x, k\rangle$ by our actions for η. (These are the ones assigned to $\langle \eta, x, k, v\rangle$ chips.) As we have argued above, these actions will be finitary. The boxes received by $\eta\widehat{\ }\langle x, k\rangle$ which are larger than those used for coding by P_j are passed on to its immediate successor $\eta\widehat{\ }\langle x, k\rangle\widehat{\ }0$ on the priority tree for its use.

Note that if $\langle x, k\rangle$ is the leftmost value of $g(\eta, s)$ taken on infinitely often, then $\eta\widehat{\ }\langle x, k\rangle$ is accessible at infinitely many C–true η–stages. (If no preceeding value is taken on for any $s \geq s_0$, η is accessible at s_0 and $g(\eta, t) = \langle x, k\rangle$ for some $t > s_0$ then $\eta\widehat{\ }\langle x, k\rangle$ is accessible at the first C–true η–state $t' \geq t$.)

Assume η is on the true path. We analyze its action to see that we can maintain our inductive assumptions about the eventual effects of α. This will allow us to complete our proof. Let $\langle x,k\rangle$ be the leftmost value taken on infinitely often by $g(\eta,s)$. The definition of accessibility guarantees that $\eta\widehat{\ }\langle x,k\rangle$ is accessible at infinitely many C–true η–stages and that, from some point s_0 on, no node to its left gets a chip or becomes accessible. Thus no node to its left can act to redefine any boxes after stage s_0. By the assignment of sequences of boxes to chips in order, it is clear that the assignment to chips to the left of $\eta\widehat{\ }\langle x,k\rangle$ is eventually constant on the C–true η–stages.

Suppose first that $k=0$ and so the answer to $Q(\eta)$ is no. In this case, all sufficiently large boxes are in state $\tau\widehat{\ }0$ at all sufficiently large C–true η–stages and are eventually all assigned to $\langle\eta,x,0,v\rangle$ chips by construction. This suffices to continue our induction.

Next suppose $k=1$ and so the answer to $Q(\eta)$ is yes. In this case, $\eta\widehat{\ }\langle x,1\rangle$ gets infinitely many chips. Each of them eventually gets a sequence of boxes $B_{j,s},\dots,B_{k,s}$ assigned at C–true η–stages which are associated with C–correct computations. Each one also eventually gets to amalgamate such a sequence when $\eta\widehat{\ }\langle x,k\rangle$ is accessible at a C–true η–stage. Thus all sufficiently large boxes are eventually assigned to chips $\langle\eta,x,1,v\rangle$ and wind up in state $\tau\widehat{\ }1$. (Remembers that if $\eta\widehat{\ }\langle x,k\rangle$ gets infinitely many chips then all boxes not assigned to chips belonging to higher priority nodes that are in e–state $\tau\widehat{\ }k$ will be eligible to be assigned to these chips belonging to $\eta\widehat{\ }\langle x,k\rangle$. If $\eta\widehat{\ }\langle x,k\rangle$ is on the true path then only finitely many boxes are assigned to chips belonging to nodes of higher priority and only finitely many are not eventually in e–state $\tau\widehat{\ }k$. Thus almost all boxes will actually be assigned to chips of the form $\langle\eta,x,k,v\rangle$.) As $\eta\widehat{\ }\langle x,1\rangle$ takes no action for a box once it and all its predecessors are in state $\tau\widehat{\ }1$, we once again have continued the inductive hypotheses.

It is now clear that every box $B_{i,s}$ is eventually constant. (It is eventually assigned to some chip assigned to a node β on or to the left of the leftmost path. After that point, it can be changed only by nodes $\gamma\leq\beta$. We have, however, shown that their outcomes are finitary.)

We can now argue that the node η on the leftmost path which is associated with R_e actually satisfies the requirement. Let $\eta\widehat{\ }\langle x,k\rangle$ be on the leftmost path and let s_0 be such that all action by nodes $\gamma<\eta\widehat{\ }\langle x,k\rangle$ has ceased and such that there is a fixed assignment of boxes to such γ on all the C–true η–stages greater than s_0.

Suppose first that $k=0$. We prove that $\Phi_e(A\oplus C)\leq_{tt} C$. As the answer to $Q(\eta)$ is no, we may also assume that after s_0 no more computations show up at C–true η–stages that would make the state of B_n be $\tau\widehat{\ }1$ for any $n>$ some fixed n_0. Assume the $B_{i,s}$ have reached their limits for $i\leq n_0$. To compute $\Phi_e(A\oplus C;x)$, find a stage s at which $\eta\widehat{\ }\langle x,0\rangle$ is accessible, $\Phi_{e,s}(A_s\oplus C_s;x)\downarrow$ and such that every $z<\varphi_{e,s}(x)$ not in a box permanently assigned to a chip belonging to some node of higher priority than $\eta\widehat{\ }\langle x,0\rangle$ is in A or a box in state $\tau\widehat{\ }0$. We claim that $\Phi_e(A\oplus C;x)=\Phi_e(A_s^{d(n_0)}\oplus C;x)=\Phi_e(A_s\oplus C_s;x)$. If not, $\Phi_e(A\oplus C;x)=\Phi_e(A_t\oplus C_t;x)$ for some C–true stage η–stage t but $A_t^{d(n_0)}=A_s^{d(n_0)}$ on the use of this computation by our choice of n_0 and s. In this case, we would have a computation at the C–true η–stage t showing that the state of B_{n_0} is $\tau\widehat{\ }1$ for a contradiction. This procedure gives a tt–reduction from C computing $\Phi_e(A \oplus C)$ as required.

Finally, suppose $k = 1$, the answer to $Q(\eta)$ is yes and $\langle x, 1\rangle$ is the leftmost outcome accessible infinitely often. In this case, we claim that $A \leq_{tt} \Phi_e(A \oplus C) \oplus C$. To compute $A(y)$, find a stage s by which all higher priority nodes have reached their final state; $\Phi_{e,s}(A \oplus C; y)\downarrow$ and all numbers $z < \varphi_{e,s}(y)$ not in boxes assigned to chips belonging to higher priority nodes are in A or a box in state $\tau\widehat{\ }1$ assigned to a chip belonging to $\eta\widehat{\ }\langle x, 1\rangle$. Now check each box in turn to see that the computation providing the witness for the high e state is C–correct. If not, then we know it, and all subsequent boxes $< \varphi_{e,s}(y)$, will go into A. If it is C–correct, we ask $\Phi_e(A \oplus C)$ which of the two associated computations gives the final answer and so determine if this block (and so all later ones $< \varphi_{e,s}(y)$) will go into A or not. Checking each box in this way using $C \restriction \varphi_{e,s}(y)$ and $\Phi_e(A \oplus C)$ at the relevant witness points, we can determine $A \restriction \varphi_{e,s}(y)$. This procedure provides the required truth table reduction and concludes the proof of (i).

Part (ii) follows immediately from (i) and Theorem 2.7. Standard Δ^0_2 approximation techniques can be used to extend the result to the tt and wtt degrees below $0'$ and so prove (iii). We omit the (many) technical details needed. □

We now know that the low$_2$ r. e. tt–degrees are precisely those with minimal covers. One might guess that they are also the tt–degrees which are minimal covers. The following result shows that this is not the case.

(3.4) Theorem. *There exists a low r. e. nonrecursive set A such that A is not a minimal cover in the r. e. tt–degrees.*

Proof: We build $A = \cup_s A_s$, together with auxiliary r. e. sets Q_e to meet the following requirements:

$$P_e : \overline{A} \neq W_e$$

$$N_e : (\exists^\infty s)(\Phi_{e,s}(A_s; e)\downarrow) \to (\Phi_e(A; e)\downarrow)$$

$$R_e : \Gamma_e(A) = V_e \Rightarrow \big[(A \leq_{tt} V_e) \text{ or } (Q_e \leq_{tt} A \ \&\ (\forall i)(\hat{R}_{e,i}))\big],$$

where

$$\hat{R}_{e,2i} : \Delta_i(V_e) \neq Q_e$$

$$\hat{R}_{e,2i+1} : \Delta_i(V_e \oplus Q_e) \neq A.$$

Here Δ_i and Γ_e list the partial tt–functionals with uses δ_i and γ_e, respectively, and $\langle V_e, \Gamma_e\rangle$ is a listing of all pairs consisting of an r. e. set and a partial tt–functional.

Once again, the construction will employ boxes and dumping. Let $\overline{A}_s = \{a_{j,s} : j \in \omega\}$. Let $A^i_s = A_s \cup \{a_{j,s} : i \leq j \leq s\}$. We ensure that if $A_{s+1} \neq A_s$ then $A_{s+1} = A^i_s$ for some i. Moreover, each node α in the construction will have at stage s, a division of $\overline{A}$ into boxes $B_{\alpha,i,s}$ with first element $b_{\alpha,i,s} = a_{d(\alpha,i,s)}$. It may process these boxes by an amalgamation procedure and then hand them on to its successor nodes. In any case, each node α enforces the rule that if any element of $B_{\alpha,i,s}$ is put ito A by a node $\beta \geq \alpha$ then all of $B_{\alpha,i,s}$ enters A. Let $\ell(e,s) = \max\{y : \forall x < y(\Gamma_{e,s}(A_s; x) = V_{e,s}(x))\}$. As usual, we regard $\Gamma_{e,s}(A_s)$ as controlling $V_{e,s}$ in the sense that once we have $\ell(e,s) > x$ we will not allow $V_{e,s}(x)$ to change unless A_s changes on $\gamma_{e,s}(x)$, the use of the computation.

A node α assigned to R_e checks for α–expansionary α–stages, i.e., $\ell(e,s)$ reaches a new maximum length of agreement on the α–stages. If s is not expansionary, α's outcome is f and no nodes extending $\alpha\widehat{\ }f$ are assigned to any $\widehat{R}_{e,j}$. If s is expansionary, α first sees if it can kill R_e off once and for all. Thus it first asks if there is some "α–active" $b_{\alpha,j,s}$ such that for some $q < \ell(e,s)$, $V_{e,s}(q) = 1$ yet $\Gamma_e(A_s^{d(\alpha,j,s)}; q) = 0$. If this is the case, we set $A_{s+1} = A_s^{d(\alpha,j,s)}$ and preserve $A \upharpoonright \gamma_e(q)$ with priority α. It also asks if there is a q such that $V_{e,s}(q) = 0$ and α–active $b_{\alpha,i,s} < b_{\alpha,j,s}$ such that $\Gamma_e(A_s^{d(\alpha,i,s)}; q) = 0$ but $\Gamma_e(A_s^{d(\alpha,j,s)}; q) = 1$. In this case, we also let $A_{s+1} = A_s^{d(\alpha,j,s)}$ and preserve the current disagreement with priority α. Should q later enter V_e at t we can set $A_{t+1} = A_t^{d(\alpha,i,t)}$ and preserve a permanent diagonalization. (Note that, by our preservation procedure at s, $d(\alpha,i,s) = d(\alpha,i,t)$ and so $A_t^{d(\alpha,i,t)} \upharpoonright s = A_s^{d(\alpha,i,s)} \upharpoonright s$. As $\gamma_e(q) < s$, our action produces the desired diagonalization.) In either of these two cases, we again have a finitary outcome for α that imposes finite restraint on A and requires no further work on R_e. (The "*α-active*" j (or $b_{\alpha,j,s}$) are those for which it is still possible (with priority α) to set $A_{s+1} = A_s^{d(\alpha,j,s)}$ i.e., there is no restraint of higher priority on $b_{\alpha,j,s}$ and they are in the e–state expected by α.)

If none of the finitary wins are possible we attempt to make $A \le_{tt} V_e$ via an "e–state amalgamation of boxes" procedure that makes V_e emulate the outcome of A. We first claim that $\Gamma_{e,s}(A_s^{d(\alpha,i+1,s)}) \subseteq \Gamma_{e,s}(A_s^{d(\alpha,i,s)})$ for each α–active i. If not, there would be a q that could be put into $\Gamma_e(A)$ by putting $b_{\alpha,i+1,s}$ into A but not by putting in $b_{\alpha,i,s}$. If $q \in V_{e,s}$, we could force the first type of finite diagonalization for R_e. If $q \notin V_{e,s}$, we could force the second. Next, note that if $v \in \Gamma_{e,s}(A_s^{d(\alpha,i,s)}) - \Gamma_{e,s}(A_s^{d(\alpha,i+1,s)})$ then $v \in \Gamma_{e,s}(A_s^{d(\alpha,j,s)})$ for all α–active $j < i$. (Otherwise, we would again be able to get a finitary win of the second type.)

Now if $\Gamma_{e,s}(A_s^{d(\alpha,i,s)}) \supsetneqq \Gamma_{e,s}(A_s^{d(\alpha,i+1,s)})$, we let $v(\alpha,i,s)$ denote the least v in the former set but not the latter (if one exists). The key point is that, if there is such a number, then $b_{\alpha,i,s} \in A$ iff $v(\alpha,i,s) \in V_e$. Our goal is to get such a number for each i and so make $A \le_{tt} V_e$. To do this we amalgamate boxes $B_{\alpha,i,s}$ to increase their e–state. More precisely, we say the state of a box $B_{\alpha,i,s}$ at e is 1 if there is a number $v(\alpha,i,s)$ as desired and 0 otherwise. Assume that the boxes $B_{\alpha,i,s}$ for $i > e$ are all in the same $(e-1)$–state (α's guess at the well resided one). If $B_{\alpha,i,s}$ is in the low e–state and some $B_{\alpha,j,s}$, $j > i$, is in the high one, we amalgamate all the boxes $B_{\alpha,i,s}, \dots, B_{\alpha,j,s}$ to produce our new box $B_{\alpha,i,s+1}$ which now enters the high e–state. (By definition $v(\alpha,j,s) \in \Gamma_{e,s}(A^{d(\alpha,j,s)}) - \Gamma_{e,s}(A^{d(\alpha,j+1,s)})$. The amalgamation makes $d(\alpha,i+1,s+1) = d(\alpha,j+1,s)$ and so $A^{d(\alpha,i+1,s+1)} \upharpoonright s = A^{d(\alpha,j+1,s)} \upharpoonright s$, $v(\alpha,j,s) \notin \Gamma_{s,s+1}(A^{d(\alpha,i+1,s+1)})$ and $\gamma_e(v(\alpha,j,s)) < s$. Of course, we are still assuming that no finite win is available and so $\Gamma_{e,s}(A_s^{d(\alpha,k+1,s)}) \subseteq \Gamma_{e,s}(A^{d(\alpha,k,s)}$ for each k with $i \le k \le j$. Thus $\Gamma_{e,s}(A_s^{d(\alpha,j,s)}) \subseteq \Gamma_{e,s}(A^{d(\alpha,i,s)})$. As our amalgamation keeps $d(\alpha,i,s) = d(\alpha,i,s+1)$, $A^{d(\alpha,i,s)} \upharpoonright s = A^{d(\alpha,i,s+1)} \upharpoonright s$ and so $v(\alpha,j,s) \in \Gamma_{e,s+1}(A^{d(\alpha,i,s+1)})$ and is the required $v(\alpha,i,s+1)$ witnessing that $B_{\alpha,i,s+1}$ in the high e–state.)

As usual the infinitary outcome that all boxes are eventually in the high e–state is the leftmost one on the tree of strategies. The outcome that almost all boxes are always in the low e–state is next and its action is finitary in nature. (The former may amalgamate boxes infinitely often, the later never does.) The truely finite outcomes producing permanent diagonalizations discussed above can be grouped as the rightmost outcome on the tree.

If α eventually gets every box $B_{\alpha,i,s}$ into the high e–state, $A \leq_{tt} V_e$: To see if $x \in A$ (for x sufficiently large) wait until x is in A or in a box $B_{\alpha,i,s}$ in the high e–state. If x is not yet in A, $x \in A$ iff $v(\alpha, i, s) \in V_e$ and no further action is needed for R_e. The last outcome to consider is that $\ell(e, s) \to \infty$, there are no finite wins for R_e and, for some i_0, the $B_{\alpha,i,s}$ boxes are all in the low e–state for $i > i_0$ and α eventually stops acting at all. In this case, we must satisfy every $\widehat{R}_{e,j}$ on the path extending α for the version Q_α of Q_e that is built below the outcome of α.

To satisfy $\widehat{R}_{e,2i}$ at some node $\beta \supseteq \alpha$ we choose the first box $B_{\beta,i,t}$ $(i > |\beta|)$ not restrained by a node of higher priority. If the last element of this box is $< t$, we amalgamate enough B_β boxes into it to make its last element be $x > t$. We designate this x as β's follower of $\widehat{R}_{e,2i}$ and guarantee that $x \in Q_\alpha$ iff $x \in A$.

As long as $\neg(\Delta_{i,s}(V_{e,s}; x)\downarrow = 0)$, we keep $x \in B_{\beta,i,s}$ out of A to meeet $\widehat{R}_{e,2i}$. If we ever get $\Delta_{i,s}(V_{e,s}; x) \downarrow= 0$, at a β–stage s, we let $A_{s+1} = A_s^{d(\beta,i,s)}$. Of course, this puts x into Q_α by our previous guarantee. The general format of our construction guarantees that (if β is on the tru

e path and we are never to its left again) that for every $t \geq s$ if $A_{t+1} \neq A_t$ then $A_{t+1} = A_t^{d(\beta,j,t)}$ for some $j \geq i$. If such a change would cause a change in $\Gamma_e(A) \restriction \delta_{i,s}(x)$ then either we could have killed off R_e by a diagonalization at α or we could have increased the e–state of some box at α contrary to our assumptions that β is on the true path and that we are never again to its left. Thus $\Delta_i(V_e; x) = 0 \neq Q_\alpha(x)$ as required to meet $\widehat{R}_{e,2i}$.

The procedure for a $\beta \supseteq \alpha$ devoted to $\widehat{R}_{e,2i+1}$ is similar albeit the argument is a bit more complicated. We again choose as a follower x the least $b_{\beta,i,s}$ not restrained with higher priority. Now we declare that $x \notin Q_\alpha$ and wait for $\Delta_{i,s}(V_{e,s} \oplus Q_{\alpha,s}; x)$ to converge.

Note that, if for no number y with $x < y < \delta_i(\Gamma_e(A) \oplus Q_\alpha; x))$ have we declared that $y \in Q_\alpha$ iff $y \in A$ (for the sake of some $\widehat{R}_{e,2p}$), then we can win $\widehat{R}_{e,2i+1}$ immediately by setting $A_{s+1} = A_s^{d(\beta,i,s)}$, since this causes x to enter A with no change to $\Delta_i(\Gamma_e(A) \oplus Q_\alpha)$. The point here is again that $\Gamma_e(A_s)$ and $\Gamma_e(A_s^{d(\beta,i,s)})$ must be identical since we are not able to diagonalize or raise states.

The more difficult case is that for some (least) number $k > d(\beta, i, s)$ we have declared that $a_{k,s} \in A$ iff $a_{k,s} \in Q_\alpha$. By our choice of i, $a_{k,s}$ is not controlled by any requirement of priority higher than β. So, the first thing β does is to see if it can meet $\widehat{R}_{e,2i+1}$ by setting $A_{s+1} = A_s^{d(\beta,i,s)}$. That is, we see if $\Delta_{i,s}(\Gamma_{e,s}(A_s^{d(\beta,i,s)}) \oplus Q^i_{\alpha,s}; x) = 0$. Here $Q^i_{\alpha,s}$ denotes what $Q_{\alpha,s+1}$ would be if we set $A_{s+1} = A_s^{d(\beta,i,s)}$. Should such a win exist, we set $A_{s+1} = A_s^{d(\beta,i,s)}$, and meet $\widehat{R}_{e,2i+1}$ by x entering A. Of course, we now impose restraint of priority β to preserve the win. Suppose that no such win exists. Then we see if we can win by setting $A_{s+1} = A_s^{d(\beta,j,s)}$ instead where j is least such that $a_{k,s} \in B_{\beta,j,s}$. That is, does $\Delta_i(\Gamma_e(A_s^{d(\beta,j,s)}) \oplus Q^j_{\alpha,s}; x) = 1$? Again, if such a win exists we take it. If neither of these are true then since $Q^i_{\alpha,s} =$

$Q^j_{\alpha,s}$, it can only be that $\Gamma_e(A_s^{d(\beta,i,s)}) \neq \Gamma_e(A_s^{d(\beta,j,s)})$ below $\delta_i(x)$. Once again, this would mean that we would have had a different outcome at α by diagonalization or state raising for a contradiction. Thus we can meet $\widehat{R}_{e,2i+1}$.

Finally, the requirements P_e and N_e are met in the usual way. A node σ acting for N_e just imposes restraint whenever $\Phi_{e,s}(A_s;e)$ converges at a σ–stage. A node τ acting for P_e chooses a follower $x = b_{\tau,i,s}$ not restrained with higher priority. If at a later τ–stage t, $x \in W_{e,t}$, we put $B_{\tau,i,s}$ $(= B_{\tau,i,t})$ into A.

Note that all positive outcomes and all restraints are finitary so all the requirements (in particular the N_e) are eventually satisfied by the actions of nodes on the true path.

The last point to be verified is that if we do not get $A \leq_{tt} V_e$ at the node α on the true path devoted to R_e, then the version Q_α of Q_e that we build below α is *tt*–reducible to A. Now the only way a number x can get into Q_α is for it to be attached to A's action at x by some $\beta \supseteq \alpha$ is acting for $R_{e,2i}$. When this assignment is made we guaranteed that $x > s$. Thus it suffices to go to stage $x+1$ and see if x has been attached to A. If not $x \notin Q_\alpha$. If so, $x \in Q_\alpha$ iff $x \in A$. $\square$

(3.5) Theorem. *(i) If* $\mathbf{c}$ *is low*$_2$ *and r. e. then there exists an incomplete* 1*–topped r. e. degree* $\mathbf{a} \geq \mathbf{c}$.

(ii) Since all incomplete 1*–topped degrees are low*$_2$ *by Jockusch [1972] (Corollary 8 (i)), it follows that* $\mathbf{c}$ *is low*$_2$ *iff* $\mathbf{c}$ *is bounded by an incomplete* $1-$ *topped-r. e. degree.*

(iii) Consequently, the class low$_2$ *is definable in the structure of r. e. sets with two degree orderings,* $\leq_T$ *and* $\leq_r$ *for* $r \in \{1, m, tt, wtt\}$. *(This uses the facts that every incomplete r. e. wtt-topped T-degree is low*$_2$ *and that the set constructed above the given low*$_2$ *degree is in fact above it in the* 1*-degrees. The first fact is an index set result proven exactly as for the* 1*–topped or* m*–topped r. e. degrees as* $\{e \mid W_e \leq_{wtt} A\}$ *is* Σ_3 *for any r. e. A.)*

Proof: Let γ_e denote the e–th partial recursive function. We need to build A and an auxiliary r. e. set B in stages to meet the following requirements.

$$Q_e : \neg(B \leq_m A \oplus C \text{ via } \gamma_e)$$
$$R_e : \Phi_e(A \oplus C) = W_e \text{ implies } W_e \leq_1 A.$$

Here $(\Phi_e, W_e)_{e\in\omega}$ is a listing of all pairs consisting of a (partial) Turing reduction and an r. e. set. We use the standard convention that $\Phi_e(A \oplus C)$ controls W_e in the sense that once a computation halts and we have a length of agreement $l(e,s) > x$ then we don't allow $W_{e,t}(x)$ to change for $t > s$ unless some number below the use $\varphi_e(x,s)$ enters either A or C.

We begin by briefly reviewing the construction of an m–topped degree (i.e. with $C = \emptyset$) from Downey and Jockusch [1987]. We will then discuss how to work above a given low C, and, finally, how to work above a low$_2$ C.

The manner in which we meet the R_e is essentially positive. Suppose node σ is devoted to R_e. We let $\ell(\sigma,s)$ be the length of agreement between $\Phi_e(A \oplus C)$ and W_e at σ–stage s. We also denote W_e by W_σ as usual. The idea is that, at the first σ–stage when $l(\sigma,s) > x$, if $x \notin W_{e,s}$ we define a coding marker $f(\sigma,x) \notin A_s$, chosen as a fresh number. We then promise that $f(\sigma,x) \in A$ iff $x \in W_e$. So if x

enters W_e, we are committed to enumerating $f(\sigma, x)$ into A. The outcome of σ is i (for infinitary) when $\ell(\sigma, s)$ reaches a new maximum and f (for finite) otherwise.

The action for a node τ devoted Q_e is as follows. We pick a follower z targeted for B and wait till $\gamma_{e,s}(z)\downarrow$. Then if $\gamma_{e,s}(z) \in A_s$ already, we win by keeping z out of B. If $\gamma_{e,s}(z) \notin A_s$, then we can win by adding z into B and keeping $\gamma_{e,s}(z)$ out of A. The outcomes for τ are all finitary. Either we wait (w) for the follower z to be realized $(\gamma_e(z)\downarrow)$ or we diagonalize (d) by putting z into B and preserving A.

(Note that another reasonable strategy is to put $\gamma_{e,s}(z)$ into A and keep z out of B. The problem with this approach is that for some pair $\langle \sigma, q\rangle$ it may be that $f(\sigma, q) = \gamma_{e,s}(z)$. So such a $\gamma_{e,s}(z)$ is forbidden to enter A unless q enters W_σ. This means that the entry of $\gamma_{e,s}(z)$ into A is more or less out of our control if we respect f's wishes. Actually, the dual problem of being forced to keep some q out of W_σ will be encountered below. The difference is that we can hope to keep numbers out of W_σ via our control of A while we have no way to put them into W_σ.)

In the basic construction (no C), the other thing we need to do is to make A nonrecursive. The corresponding requirements acting, for example, at a node β add elements into A for diagonalization initiating a sequence of coding actions since some x entering A can allow $W_{e,s}$ to change for many e. This action in turn puts many $f(\sigma, y)$ into A. The principal task in any one–topped construction is to mesh the interaction of such codings with the satisfaction of the Q_e: We must know that, when we add z into B for β, all pending coding commitments (of higher priority) have ceased to act. (After all, $f(\sigma, y) = \gamma_{e,s}(z)$ is entirely possible; if y enters W_σ after we put z into B, we are committed to putting $f(\sigma, y)$ into A and hence $\gamma_{e,s}(z)$ into A. Now we would have no contradiction since $z \in B$ and $\gamma_{e,s}(z) \in A$).

We can, however, actively prevent such y from entering W_σ by preserving A on $\varphi_e(y)$. We can do this simultaneously for all possible actions of higher priority R requirements (say at node $\alpha \subseteq \beta$) with infinitary outcomes contained in β as long as all the higher prior ity Q requirements have ceased to act. (This is eventually guaranteed by the usual tree mechanisms.) The point is that, if $\alpha\widehat{\ }i \subseteq \beta$, then we act for β only at a α–expansionary stages, s, when its length of agreement function is at a new maximum. Thus preserving $A \upharpoonright s$ will preserve W at every x with a coding marker already appointed. Of course, if $\alpha\widehat{\ }f \subseteq \beta$ we may eventually assume that no further action is taken for the sake of α.

We now turn to the case where C is low. Note that we now no longer need to make A nonrecursive. The principal difficulty will be to make $A \oplus C$ not 1–complete.

Now if C is low then $C' \equiv_T \emptyset'$. Hence, in a standard way via the recursion theorem, test sets, and the limit lemma, we can approximate answers to C'–questions about the construction recursively in $\emptyset'$. We always reconcile the answer given by the recursive approximation to any C' question with the current apparent answer by continuing to run the enumeration of C and the recursive approximation until they agree. (We assume that the reader is familiar with this technique sometimes referred to as the "Robinson trick" (see Soare [1987, Ch XI]). We remind the reader that this typically involves a low set C and a reduction $\Delta(C)$. One asks if $\Delta_s(C_s; q)$, say, is correct by seeing if C is correct on the use of the computation $\delta = \delta_s(q)$. To answer this question, one enumerates a canonical index u for $\{z : z \leq \delta \ \& \ z \notin C_s\}$ into a "test set" V whose index i is given by the recursion theorem. Since C is low $\mathbf{0}'$ can tell if there is a $u \in W_i = V$ such that $D_a \cap C = \emptyset$. Thus we can approximate the answer as $\lim_s g(i,s)$ for some recursive g. After we enumerate u into V we reconcile any apparent inconsistencies by waiting for a stage $t > s$ where either $g(i,t) = 1$ or the Δ_s computation becomes C–incorrect since $C_s \restriction \delta \neq C \restriction \delta$. We will assume this procedure to be known and only refer to it informally in the discussion below.)

A requirement τ devoted to diagonalizing against the potential reduction γ_τ wants to get a stage s at which there is an x with $\gamma_\tau(x) \notin A \oplus C$ at which there are C–correct computations for all coding markers of priority σ such that $\sigma\hat{\ }i \subset \tau$ (i for infinitary outcome) which are still on the board. If τ can get such stage, it can put x into B and preserve A (up to the stage) as before so as to win its requirement. (By preserving A, τ guarantees that $\gamma_\tau(x)$ will never have to be put into A.) Of course, if $\gamma_\tau(x) \in A \oplus C$ we just keep x out of B for an easy win.

The problem is how to produce such a stage (if τ needs one, i.e. $\gamma_\tau(x) \notin A \oplus C$ for all the potential witnesses x). The solution is to ask the right questions (of the recursive approximation to C').

When we hit the first coding node σ we first ask the oracle approximation if there is at least one stage s at which σ is accessible and we have C–correct computations for all current σ coding markers. If the answer is no, the outcome of σ is f (for finite win). In this case, we proceed as if $\Phi_\sigma(A \oplus C)$ is not total or is not equal to W_σ, i.e. no action is necessary. In particular, no chips are issued to appoint new coding markers. (We will only appoint new coding markers when additional chips have been assigned to the node.) If the answer is yes, we see if the current answer, as judged by C_s, is that s is the first such stage. If so, the outcome is i and we issue a chip for the appointment of a new coding marker at a later stage. (Coding markers are appointed when there is a new length of agreement between $\Phi_\sigma(A \oplus C)$ and W_σ and there is an available chip to use up for the appointment.) If s is not the first such stage (according to C_s), we ask (the oracle approximation) if there are at least two such stages in the construction. If the answer is no, the outcome of σ is f. If yes, we check if s now seems to be the second such stage. If so the outcome is i (and we issue a chip). If not, we ask if there are at least three such stages, etc. Eventually, we either get an outcome f or s seems to be the nth such stage and we get an outcome i.

If there is a question in this sequence whose final answer is no, we argue that no action is really needed for σ. Let $n+1$ be the first such, i.e. there are n stages as described but not $n+1$ many such stages. Once we are beyond the n correct stages and the answers to the first $n+1$ questions have reached a limit, no new chips can be issued and so eventually no new markers can be appointed. Thus there is never a stage at which there are C–correct computations for all the finitely many markers on the board. It follows that it cannot be the case that $\Phi_\sigma(A \oplus C) = W_\sigma$ and the overall effect of σ is finite. The construction can easily live with this.

On the other hand, if every question has a final answer yes, then we have outcome i infinitely often at stages at which the computations are really C–correct. (Consider the first such s. It appears C–correct at stage s, of course, and we run the approximation until it gives the answer that there is such a stage.)

The next coding requirement α below $\sigma\hat{\ }i$ asks for stages at which all the current σ computations are C–correct and if, among such stages, there are n which are also C–correct on all the α markers. If we really have to worry about α, then there will be infinitely many stages at which $\alpha\hat{\ }i$ is accessible and all the σ and α computations are C–correct.

Eventually we get to τ which is working on requirement Q_τ. If we have to worry about τ there will eventually be a stage at which τ is accessible, all the σ, α etc. computations are C–correct, and there is a witness x with $\gamma_\tau(x) \notin C$. At such a stage we can preserve A and win τ. (Of course, we are asking if there is such a stage from the recursive approximation. If it answers no we need do nothing for τ. If yes, we put x into B and try to preserve A (until C changes when we start again).)

At the end of every stage we put into A every pending coding marker (i.e. ones appointed to code the fact that "x is in W" when x has in fact gone into W) for which we have a computation from $A \oplus C$ saying it is in W and which is not restrained by any requirement of higher priority.

We will now describe how to perform the construction above a $\overline{low_2}$ r. e. set C. The idea is to replace the above series of questions by one: Are there infinitely many σ–stages where the current σ markers have C–correct computations. Note that this question is Π_2^C and hence Δ_3 as C is low$_2$. We shall approximate the answers as in Shore–Slaman [1992] and the proof of Theorem 3.3. Our construction is more involved and technically more difficult as it has additional coordination problems.

We remark that coding markers will be issued not just for σ but for each positive outcome (of the Δ_3 approximation) for σ when the outcome has a chip and seems correct. They will be cancelled when nodes to the left become accessible and put in only when the outcome with which they are associated are accessible. The crucial point in the construction is that a node associated with the diagonalization requirement for γ_τ needs a certain type of configuration to be able to win. We describe the required situation in the following definition.

(3.6)Definition. *Let η be a node on the tree of outcomes. Then we say that s is η–closed at t if $(\forall z < s)\forall\sigma, x, y\{[(z$ is a $\sigma\hat{\ }\langle x,1\rangle$–marker for $y)$ & $(\sigma\hat{\ }\langle x,1\rangle \subseteq \eta)] \rightarrow y < \ell(\sigma,t)$ & $\varphi_{\sigma,t}(A_t \oplus C_t; y) < s]$ & $[(z$ is an $\eta\hat{\ }\langle x,1\rangle$-marker for $y \rightarrow y < \ell(\eta,t)$ & $\varphi_{\eta,t}(A_t \oplus C_t; y) < s)]\}$. (The $\eta\hat{\ }\langle x,1\rangle$–markers relevant in this definition are the ones still defined when some immediate successor of η has been declared accessible. The computations are done with the oracles as they exist when η is declared accessible and ℓ is the length of agreement function appropriate to the requirement.)*

We say that s is C–correct at t if $C\restriction s = C_t\restriction s$.

The idea here is that if τ is a diagonalization node for Q_τ and there are $x < s < t$ such that, at t, τ is accessible; $\gamma_\tau(x) < s$ and is not in A; and s is τ–closed and C–correct, then we can win the τ requirement at t by putting x into B and preserving $A\restriction s$ against all lower priority requirements. Thus this is the situation we are trying to produce.

We now describe the construction assuming familiarity with the methods used in Theorem 3.3 as the general format of this construction is the same.

Construction

As usual, whenever a node α becomes accessible, we initialize all nodes to its right by canceling all chips, markers and restraints associated with them. Suppose we are at stage t. We describe our actions based on the type of requirement assigned to the node that has just become accessible. Suppose $\sigma = \eta\widehat{\ }\langle y, j\rangle\widehat{\ }r$ is associated with the coding requirement for $\Phi_\sigma(A \oplus C) = W_\sigma$. (The "$r$" at the end of the sequence is the restraint imposed by the action associated with $\eta\widehat{\ }\langle y, j\rangle$.) The question whose answer we approximate in our Δ_3 way at σ is

$Q(\sigma) : \forall n \exists s > n \exists t > s[\sigma$ is accessible during stage t and, when some immediate successor of σ is declared accessible at t, s is σ–closed and C–correct].

At stage t, we assign a chip $\langle \sigma, x, k, t\rangle$ to the appropriate outcomes $\langle x, k\rangle$ of σ (as determined by our approximation to the answer to $Q(\sigma)$) and determine the accessible immediate successor of σ as in the construction for Theorem 3.3 and so, in particular, initialize all nodes to the right of this immediate successor. (Thus any outcome to the left of the true one gets only finitely many chips, appoints only finitely many markers and is accessible only finitely often. Moreover, if $\langle y, j\rangle$ is the true outcome, it is accessible infinitely often and indeed at infinitely many C–true σ–stages.) We next list the stages $s_1, \ldots, s_n$ which are $\sigma\widehat{\ }\langle x, 1\rangle$ closed (and surely appear C–correct as we can only use C_t to check) at t and assign them in order to the still current $\sigma\widehat{\ }\langle x, 1\rangle$ chips for each outcome $\langle x, 1\rangle$ of σ. At t, each $\sigma\widehat{\ }\langle x, k\rangle$ imposes restraint $r(\sigma\widehat{\ }\langle x, k\rangle, t)$ equal to the largest s so assigned to any chip assigned to a node to its left. We also assign a new large number as a $\sigma\widehat{\ }\langle x, 1\rangle$ coding marker for each z less than both $\ell(\sigma, t)$ (the associated length of agreement) and the number of chips assigned to $\sigma\widehat{\ }\langle x, 1\rangle$ which does not yet have such a $\sigma\widehat{\ }\langle x, 1\rangle$ coding marker. If the accessible outcome for σ is $\langle x, 1\rangle$, we put into A all $\sigma\widehat{\ }\langle x, 1\rangle$ coding markers that are not restrained with priority at least that of $\sigma\widehat{\ }\langle x, 1\rangle$ whose corresponding number is now in W_τ and less than the current length of agreement for the associated functional. In any case, the outcome for the accessible immediate successor $\sigma\widehat{\ }\langle x, k\rangle$ of σ is $\sigma\widehat{\ }\langle x, k\rangle\widehat{\ }r(\sigma\widehat{\ }\langle x, k\rangle, t)$.

We now consider a node $\mu = \sigma\widehat{\ }\langle z, j\rangle\widehat{\ }r$ associated with the diagonalization requirement that γ_μ does not reduce B to A that has just become accessible. The node μ wants a z such that $\gamma_{\mu,t}(z) \downarrow\notin A_t \oplus C_t$ and s, $t > z, \gamma_u(z)$ such that, at t, μ is accessible, s is μ-closed and C–correct. At such a stage t, μ would like to put z into B and preserve $A \restriction s$ against all lower priority requirements for a win. We ask the question:

$$\begin{aligned} Q(\mu) : \forall n \exists z, s, t > n[(\gamma_{\mu,t}(z) \downarrow\notin A_t \oplus C_t) \;\&\; (z, \gamma_\mu(z) < s < t) \text{ and, when} \\ \mu \text{ is accessible during stage } t,\ s \text{ is } \mu\text{–closed and } C\text{–correct}] \end{aligned}$$

We approximate the answer to this question, assign chips to outcomes $\langle x, k\rangle$ and determine the accessible immediate successor for μ as before. Each chip assigned to a successor of μ wants to have assigned to it a number z which will be a potential diagonalization witness with certain properties as well as a stage s which will include the restraint it needs to impose to preserve the diagonalization. A chip assigned to $\langle x, k\rangle$ wants to be assigned, at a stage t when μ is accessible, a $z \in \omega^{(\mu,x,k)}$ which is larger than each z' assigned to any smaller chip belonging to $\langle x, k\rangle$ as well as an $s > z, \gamma_\mu(z)$ which, when μ is accessible at t, is μ-closed and C–correct. A chip for $\langle x, 1\rangle$, in addition, wants $\gamma_\mu(z)\downarrow\notin A \oplus C_t$. We assign numbers z and s that satisfy the desires of the outcomes of μ to the chips assigned to all these outcomes in order of the priority of the chips as long as there are such numbers available. If $\langle x, 0\rangle$ is the accessible outcome of μ, we take no action for γ_μ and it imposes no restraint. Thus the next accessible node is just $\mu\widehat{\ }\langle x, 0\rangle$ followed by 0. If $\langle x, 1\rangle$ is the accessible outcome of μ, we put into B any z assigned to one of its chips and restrain $A\restriction s$, with priority $\mu\widehat{\ }\langle x, 1\rangle$ for the associated s unless there is a smaller z for which $\mu\widehat{\ }\langle x, k\rangle$ is already preserving such a set up which also seems C–correct. The outcome for $\mu\widehat{\ }\langle x, k\rangle$ is this restraint that it imposes.

When we reach a node of length t, we terminate stage t.

Verifications:

Let T be the true path in the priority tree, i. e. the leftmost nodes that are accessible infinitely often. We show that each requirement is satisfied by the action of the nodes on T assigned to it. First a preliminary lemma.

(3.7) Lemma. *Suppose $\sigma \in T$ is assigned to a coding requirement and no node to the left of $\sigma\widehat{\ }\langle x, k\rangle$ is accessible after stage t_0. If, at stage $t > t_0$, a $\sigma\widehat{\ }\langle x, k\rangle$ chip is assigned a number s which is actually C–correct then s remains $\sigma\widehat{\ }\langle x, k\rangle$ closed and C–correct until $\sigma\widehat{\ }\langle x, k\rangle$ is accessible.*

Proof: By assumption $C\restriction s$ never changes. We claim $A\restriction s$ also does not change. No node to the right of $\sigma\widehat{\ }\langle x, k\rangle$ can put any number $< s$ into A because of the restraint imposed by the assignment of s to a $\sigma\widehat{\ }\langle x, k\rangle$ chip. No node to its left can put in any number without becoming accessible and so contradicting our hypothesis. Of course, no node extending $\sigma\widehat{\ }\langle x, k\rangle$ can put in any number without $\sigma\widehat{\ }\langle x, k\rangle$ first becoming accessible. If any node ρ with $\rho\widehat{\ }\langle w, 1\rangle \subset \sigma$ associated with a coding requirement wants to put a number $z < s$ into A at some t it does so because it is a $\rho\widehat{\ }\langle w, 1\rangle$ marker for some $y < s$ and y has entered W_ρ since s but z is not in A. In this case, the current length of agreement $\ell(\rho, t)$ is at most y since it was larger than y at s, neither A nor C has changed on the use since s but y has entered W_ρ. Thus we would not actually try to put z into A. $\square$

(3.8) Lemma. *i) If $\eta \in T$ and η is not assigned a coding requirement then*

$$(\star) \qquad \forall n \exists s > n \exists t > s[\text{at stage } t, \eta \text{ is accessible and, when it is accessible, } s \text{ is } \eta\text{-closed and } C\text{–correct}].$$

ii) If $\sigma \in T$ is assigned a coding requirement R_σ then our actions for its true outcome satisfy R_σ. In particular, if the true outcome of σ is $\langle x, 0\rangle$ then $\Phi_\sigma(A\oplus C) \neq W_\sigma$; if it is $\langle x, 1\rangle$ and $\Phi_\sigma(A \oplus C) = W_\sigma$ then the $\sigma\widehat{\ }\langle x, 1\rangle$ coding markers supply a $1-1$ reduction computing W_σ from A. Moreover, the restraint imposed by the true outcome $\sigma\widehat{\ }\langle x, k\rangle$ eventually attains its $\liminf$ *on almost all the C–true $\sigma\widehat{\ }\langle x, k\rangle$ stages and so it has a leftmost outcome r.*

iii) If $\mu \in T$ is assigned a diagonalization requirement Q_μ, then our actions for its true outcome satisfy Q_μ. Moreover, the restraint imposed by its true outcome $\langle x, k\rangle$ is eventually constant on the $\mu\widehat{\ }\langle x, k\rangle$ stages and so it has a leftmost outcome r.

Proof: We proceed by simultaneous induction along T by blocks (of length two) corresponding to nodes assigned to requirements and their outcomes We first consider a node $\sigma \in T$ assigned to a coding requirement R_σ with true outcome $\langle x, k\rangle$. Let t_0 be such that no node to the left of $\sigma\widehat{\ }\langle x, k\rangle$ ever gets a chip or is accessible after t_0. First, there are finitely many chips assigned to outcomes to the left of $\langle x, k\rangle$. By the Lemma above any number assigned to one of them at any sufficiently large C-true σ–stage remains assigned to it forever. Thus the set of number assigned to outcomes to the left of $\langle x, k\rangle$ is eventually constant on the C–true σ-stages and so on the subset of those stages which are also C–true $\sigma\widehat{\ }\langle x, k\rangle$ stages. Thus the restraint imposed by the true outcome $\sigma\widehat{\ }\langle x, 1\rangle$ eventually attains its $\liminf$ on almost all the C–true $\sigma\widehat{\ }\langle x, 1\rangle$ stages and so it has a leftmost outcome r as required.

Of course, for σ itself, (i) is vacuously true. We show that R_σ is satisfied and that we can continue our induction to its immediate successor $\sigma\widehat{\ }\langle x, k\rangle$ on T (and so to the end of its block) by showing that

$$\begin{aligned}\forall n \exists s > n \exists t > s[&\text{at stage } t, \sigma\widehat{\ }\langle x, k\rangle \text{ is accessible and, when it is accessible at } t,\\ &s \text{ is } \sigma\widehat{\ }\langle x, k\rangle \text{ –closed and } C\text{–correct}].\end{aligned}$$

Fix n and assume that, by $t_1 > t_0$, outcome $\langle x, k\rangle$ has more than n many chips.

a) $k = 0$**.** By our induction assumption for (i), we may choose an $s_1 > t_1$ and $t > s_1$ such that σ is accessible at t and s_1 is $\eta\widehat{\ }\langle y, j\rangle$ closed and C–correct when $\nu\widehat{\ } langley, j\rangle$ is accessible at t where $\sigma = \eta\widehat{\ }\langle y, j\rangle\widehat{\ } r$ for some r. Now any $s \le s_1$ which is $\eta\widehat{\ }\langle y, j\rangle$ closed at t is also C–correct. By our assumptions, there is at least one such $s > n$ which is assigned to a $\sigma\widehat{\ }\langle x, 0\rangle$ chip since there are more than n chips and $\eta\widehat{\ }\langle y, j\rangle$ closed is the same as $\sigma\widehat{\ }\langle x, 0\rangle$ closed. If $t' \ge t$ is the next stage at which $\sigma\widehat{\ }\langle x, 0\rangle$ is accessible, the above lemma guarantees that this $s > n$ will still be $\sigma\widehat{\ }\langle x, 0\rangle$ closed and, of course, C–correct. Thus we have established (i) for $\sigma\widehat{\ }\langle x, 0\rangle$.

We now prove that R_σ is satisfied if $k = 0$.

Claim: $\Phi_\sigma(A \oplus C) \ne W_\sigma$.

Proof (by contradiction). There is a fixed finite set of σ markers which are the only ones current when $\sigma\widehat{\ }\langle x, 0\rangle$ is accessible at all sufficiently large $\sigma\widehat{\ }\langle x, 0\rangle$ stages (as nodes to its left get only finitely many chips and so markers while ones to its right are canceled when it becomes accessible). If $\Phi_\sigma(A \oplus C) = W_\sigma$, then $\ell(\sigma, t)$ would eventually be greater than all the numbers which have σ markers at such stages and indeed via C–correct (and even A–correct) computations. Thus each sufficiently large number the additional requirements for s to be σ closed (and C–correct) over those for its predecessor $\eta\widehat{\ }\langle y, j\rangle$ will be satisfied at all sufficiently large $\sigma\widehat{\ }\langle x, 0\rangle$ stages. This contradicts the correctness of the answer to our question $Q(\sigma)$ given by the approximation. □

b) $k = 1$. By the correctness of our approximation, there is an $s > t_1 > n$ and a $t > s$ such that, when some $\sigma\widehat{\ }\langle z, j\rangle$ is accessible at t, s is σ closed and C–correct. Any such s is $\sigma\widehat{\ }\langle x, 1\rangle$ closed at the point during stage t at which σ is accessible since this actually imposes fewer demands as $\sigma\widehat{\ }\langle y, 1\rangle$ markers for $y > x$ can be ignored and, by the choice of t_0, no outcome to the left of $\langle x, 1\rangle$ can be accessible at t and so no markers relevant to the definition of $\sigma\widehat{\ }\langle x, 1\rangle$–closed can be canceled after σ is accessible. Moreover, any such s' less than s which appears C–correct is, of course, C–correct. As $\sigma\widehat{\ }\langle x,1\rangle$ has more than n chips, one such $s>n$ is assigned to one of its chips. By the lemma, it remains $\sigma\widehat{\ }\langle x,1\rangle$ closed (and C–correct) until $\sigma\widehat{\ }\langle x,1\rangle$ is accessible and so we have established (i) for $\sigma\widehat{\ }\langle x,1\rangle$.

Finally, note that $\sigma\widehat{\ }\langle x,1\rangle$ is accessible infinitely often at C–true $\sigma\widehat{\ }\langle x,1\rangle$ stages at which the restraint imposed by nodes of higher priority are eventually constant by induction. Thus almost all its markers that are ever appointed succeed in coding W_σ. If $\Phi_\sigma(A\oplus C)=W_\sigma$ then the length of agreement goes to infinity and so $\sigma\widehat{\ }\langle x,1\rangle$ coding markers are appointed for every y and so R_σ is satisfied as required.

We now turn to the diagonalization requirements. There is nothing to prove for (i) for a node $\mu=\sigma\widehat{\ }\langle z,j\rangle\widehat{\ }r$ associated with the diagonalization requirement that γ_μ does not reduce B to A or for its true successor, $\mu\widehat{\ }\langle x,k\rangle$, as μ-closed and $\mu\widehat{\ }\langle x,k\rangle$-closed are the same as $\sigma\widehat{\ }\langle z,j\rangle$–closed. We therefore prove (iii) for μ. Remember that, by induction, there are infinitely many s with stages t at which μ is accessible and s is μ–closed and C–correct.

a) $k=0$**.** So the answer to $Q(\mu)$ is "no" and there is then a stage t_0 after which no outcome to the left of $\langle x,0\rangle$ is ever accessible or assigned a chip. Now no number $z\in\omega^{(\mu,x,0)}$ is ever put into B by construction. Thus we satisfy the requirement Q_μ unless $\gamma_\mu(z)\downarrow\notin A\oplus C$ for every $z\in\omega^{(\mu,x,0)}$ so we assume this to be the case. The outcome $\langle x,0\rangle$ gets infinitely many chips and, by construction and our assumption, each one is eventually assigned a μ–closed s and a number z which is never assigned to any chip for another outcome such that $\gamma_\mu(z)\downarrow\notin A\oplus C$. By the induction hypothesis, there are then infinitely many s, t as required to make $Q(\mu)$ true for our contradiction. Thus if the true outcome is $\langle x,0\rangle$ we satisfy the requirement associated with μ. Of course, the accessible outcome for $\mu\widehat{\ }\langle x,k\rangle$ and the restraint it imposes is always 0.

b) $k=1$. So the answer to $Q(\mu)$ is "yes" and there is then a stage t_0 after which no outcome to the left of $\langle x,0\rangle$ is ever accessible or assigned a chip. The outcome $\langle x,0\rangle$ gets infinitely many chips and, by our case assumption and the rules of the construction, each one is eventually assigned, at a μ-stage t, a μ–closed C–correct s and a number $z<s$ which is never assigned to any chip for another outcome such that $\gamma_\mu(z)\downarrow\notin A_t\oplus C_t$ & $z,\gamma_\mu(z)<s<t$. Once such assignment is made after t_0 we would put z into B and preserve $A\restriction s$ unless we already have a smaller such z in B. In any case, as one is really C-correct at t, the smallest one that seems C-correct really is. We now restrain $A\restriction s$ for the appropriate s with priority $\mu\widehat{\ }\langle x,1\rangle$. By our choice of t_0 this restraint is never violated and so we satisfy Q_μ and from now on the outcome of $\mu\widehat{\ }\langle x,1\rangle$ and so the restraint it imposes is always s. □

References

[1973] Cooper, S. B., *Minimal degrees and the jump operator*, J. Symb. Logic **38**, 249–271.

[1993] Cooper, S. B., *Rigidity and definability in the noncomputable universe* (to appear).

[1973] Degtev, A. N., *tt– and m–degrees*, Algebra and Logic **12**, 78–89.

[1978] Degtev, A. N., *Three theorems on tt–degrees*, Algebra and Logic **17**, 187–194.

[1993] Downey, R. G., *Array nonrecursive sets and lattice embeddings of the diamond*, Illinois J. Math. J. **37**, 349–374.

[1989] Downey, R. G., *Recursively enumerable m– and tt–degrees I: the quantity of m–degrees*, J. Symb. Logic **54**, no. 2, 553–567.

[1990] Downey, R. G., *Lattice nonembeddings and initial segments of the recursively enumerable degrees*, Ann. Pure and Appl. Logic **49**, 97-119.

[1987] Downey, R. G. and C. Jockusch, Jr., *T–degree, jump classes and strong reducibilities*, Trans. Amer. Math. Soc. **301**, 103–136.

[1990] Downey, R. G., C. Jockusch, and M. Stob, *Array nonrecursive sets and multiple permitting arguments*, in *Recursion Theory Week* (K. Ambos–Spies, G. Müller and G. Sacks, eds.), Springer–Verlag, Berlin, LNMS **1432**, pp. 141–171.

[1995] Downey, R. G. and R. A. Shore, *Lattice embeddings below nonlow$_2$ recursively enumerable degrees*, Israel J. Math. (to appear).

[1989] Downey, R.G. and T. Slaman, *Completely mitotic recursively enumerable degrees*, Ann. Pure and Appl. Logic **41**, 119–152.

[1981] Epstein, R.L., R. Haas, and R. Kramer, *Hierarchies of sets and degrees below* $\mathbf{0}'$, in *Logic Year 1979–1980* (M. Lerman, J. Schmerl, and R. Soare, eds.), LNMS **859**, Springer–Verlag, Berlin, 32-48.

[1989] Fejer, P. J. and R. A. Shore, *A direct construction of a minimal tt–degree*, in *Recursion Theory Week* (K. Ambos–Spies, G. Müller and G. Sacks, eds.), Springer–Verlag, Berlin, LNMS **1432**, pp. 187–204.

[1972] Jockusch, C. G. Jr., *Degrees in which the recursive sets are uniformly recursive*, Can. J. Math. **24**, 1092–1099.

[1978] Jockusch, C.G. Jr. and D. Posner, *Double jumps of minimal degrees*, J. Symb. Logic **43**, 715–724.

[1972] Jockusch, C.G. Jr. and R.I. Soare, Π^0_1 *classes and degrees of theories*, Trans. Amer. Math. Soc. **173**, 33–56.

[1979] Kobzev, G., *On tt–degrees of r. e. T–degrees*, Mathematics of USSR Sbornik **35**, 1973–180.

[1968] Lachlan, A., *On the lattice of recursively enumerable sets*, Trans. Amer. Math. Soc. **130**, 1–37.

[1968a] Lachlan, A., *Degrees of recursively enumerable sets which have no maximal supersets*, J. Symb. Logic **38**, 431–443.

[1975] Ladner, R. E., *On the structure of polynomial–time reducibility*, J.ACM **22**, 155–171.

[1983] Lerman, M., *Degrees of Unsolvability*, Springer–Verlag, Berlin.

[1982] Maass, W., *Recursively enumerable generic sets*, J. Symb. Logic **47**, 809–823.

[1966] Martin, D., *Classes of recursively enumerable sets and degrees of unsolvability*, Z. Math. Logic. Grund. Math. **12**, 295–310.

[1970] Martin, D. and M. Pour.–El, *Axiomatizable theories with few axiomatizable extensions*, J. Symb. Logic **35**, 205–209.

[1980a] Nerode, A. and Shore, R. A., *Second order logic and first order theories of reducibility orderings*, in *The Kleene Symposium* (K. J. Barwise et al., eds.), North–Holland, Amsterdam, 181–200.

[1980b] Nerode, A. and Shore, R. A., *Reducibility orderings: Theories, definability and automorphisms*, Ann. Math. Logic **18**, 61–89.

[ta] Nies, A. and Shore, R. A., *Interpreting true arithmetic in the theory of the tt and wtt degrees below* $\mathbf{0}'$, in preparation.

[1981] Odifreddi, P., *Strong reduciblities*, Bull. Amer. Math. Soc. **4**, 37–86.

[1989] Odifreddi, P., *Classical Recursion Theory*, North–Holland, Amsterdam.

[ta] Odifreddi, P., *Classical Recursion Theory*, vol. 2, North–Holland, Amsterdam.

[1976] Shoenfield, J., *Degrees of classes of r. e. sets*, J. Symb. Logic **41**, 695–696.

[1985] Shore, R. A., *The structure of the degrees of unsolvability*, in *Recursion Theory*, Proc. Symp. Pure Math. **42**, (A. Nerode and R. A. Shore, eds.), AMS, Providence, RI, 33–51.

[1990] Shore, R.A. and T.A. Slaman, *Working below a low$_2$ recursively enumerable degree*, Archive for Math. Logic **29**, 201-211.

[1993] Shore, R.A. and T.A. Slaman, *Working below a high recursively enumerable degree*, J. Symb. Logic **58**, 824–859.

[1996] Slaman, T. and Woodin, H., *Definability in Degree Structures*, in preparation.

[1987] Soare, R. I., *Recursively Enumerable Sets and Degrees*, Springer–Verlag, Berlin.

[1969] Yates, C. E. M., *On the degrees of index sets II*, Trans. AMS **135**, 249–266.