ILLINOIS JOURNAL OF MATHEMATICS
Volume 37, Number 3, Fall 1993

ARRAY NONRECURSIVE DEGREES AND LATTICE EMBEDDINGS OF THE DIAMOND

ROD DOWNEY[1]

1. Introduction

In [16], Slaman solved Barry Cooper's diamond problem by constructing an r.e. degree $\mathbf{a} \neq \mathbf{0}$ that was not the top of a diamond in the Δ_2 degrees. That is, for all Δ_2 degrees $\mathbf{b}, \mathbf{c}$, if $\mathbf{b} \cup \mathbf{c} = \mathbf{a}$ and $\mathbf{b}|\mathbf{c}$ then $\mathbf{b} \cap \mathbf{c} \neq \mathbf{0}$. The proof of this result was a very complex argument involving "three jumps" of nonuniformity. Earlier work of Cooper [3], Posner [15] and Epstein [9] showed that such $\mathbf{a}$ could not be high, indeed if $\mathbf{b}$ is high then the degrees $\leq \mathbf{b}$ are complemented.

Naturally the question arises: exactly what r.e. degrees are tops of diamonds? Fejer [10] has proved that if $\mathbf{a}$ is an r.e. degree that is not the top of a diamond then $\mathbf{a}$ is low$_2$. (In fact, Fejer showed that if $\mathbf{a}$ is a degree that is non-GL$_2$ then $\mathbf{a}$ is the top of a diamond.) Subsequently, Slaman pointed out a possible definition of "low$_2$ r.e.": perhaps all low$_2$ r.e. degrees are bounded by degrees that are not tops of diamonds in the Δ_2 degrees.

The first goal of the present paper is to answer Slaman's question negatively. In fact a consequence of our results is that there are low r.e. degrees not bounded by any r.e. degree that is not the top of a diamond.

In [6], Downey, Jockusch and Stob initiated the study of a new class of r.e. degrees called the *array nonrecursive r.e. degrees*. We review the definition of this class in §2, but for our purposes here it suffices to remark that it is a natural class of degrees which arise from arguments which need 'multiple permissions'. In [6] it is shown that this class contains members of low degree, is closed upwards, and contains all non-low$_2$ r.e. degrees.

Our result is:

(1.1) THEOREM. *Suppose* $\mathbf{a}$ *is array nonrecursive. Then* $\mathbf{a}$ *is the top of a diamond in the* Δ_2 *degrees.*

Received August 10, 1988.
1991 Mathematics Subject Classification. Primary 03D25; Secondary 03D30.
[1]Research partially supported by a US/NZ cooperative grant.
The author wishes to thank Carl Jockusch and Mike Stob for fruitful discussions regarding array nonrecrusive degrees, and to thank Ted Slaman for similar discussions on the diamond problem.

Since all non-low$_2$ degrees are array nonrecursive, (1.1) subsumes Fejer's [10] result for r.e. degrees.

The proof of Theorem (1.1) is slightly reminiscent of Downey [4] where it is shown that $\mathbf{0}'$ is the top of a diamond in the d.r.e. degrees. Here the reader should recall that A is called d.r.e. if, for some r.e. $B, C, A = B - C$.

Theorem (1.1) also allows us to solve a question from [4]:

(1.2) COROLLARY. *There exist r.e. degrees* $\mathbf{a}$ *that are tops of diamonds in the* Δ_2 *degrees but not in the d.r.e. degrees.*

The proof of (1.2) depends on a slight modification of a result of Lachlan [11] and the well known fact that all nonzero d.r.e. degrees bound nonzero r.e. degrees. At this stage we should point out that—as all array nonrecursive r.e. degrees bound nonzero array recursive degrees—it follows by (e.g.) Lachlan [11] that there are degrees that are tops of diamonds in the r.e. degrees that are array *recursive*. In fact by Ambos-Spies [1] or Downey-Welch [8], there are initial segments of the r.e. degrees consisting entirely of r.e. degrees that are tops of diamonds (in the r.e. degrees). As a companion result to theorem (1.1) we shall construct various r.e. degrees $\mathbf{b}$ such that for all $\mathbf{a} \geq \mathbf{b}$ or $\mathbf{a} \leq \mathbf{b}$, $\mathbf{a}$ is the top of a diamond in the Δ_2 degrees.

The technique we use in solving Slaman's question—or rather, the technique with which we apply the array nonrecursiveness of $\mathbf{a}$ in (1.1)—has other uses. In particular we can use it to solve a question from [6]. All array recursive degrees constructed in [6] were low and it was an open question whether all array recursive degrees were low. We shall prove:

(1.3) THEOREM. *There exist low$_2$-low array recursive degrees.*

The proof of (1.3) involves a couple of intermediate steps. First we define an r.e. set A to be *W-mitotic* if there exist r.e. sets B and C splitting A (i.e. $A = B \cup C$ and $B \cap C = \emptyset$) such that $A \equiv_W B \equiv_W C$ where $\leq_W$ denotes *weak truth table* reducibility. Using the technique of (1.1) we can show:

(1.4) THEOREM. *Suppose* $\mathbf{a}$ *is array nonrecursive. Then* $\mathbf{a}$ *contains an r.e. set A which is non-W-mitotic.*

Once we have (1.4) we can then get (1.3) using a result of Ladner [13] where it is established (see Ambos-Spies & Fejer [2] and Downey-Slaman [7]) that there exists a low$_2$-low degree $\mathbf{a}$ that contains a single r.e. W-degree (i.e., a *contiguous* degree) and such that all of the r.e. elements of $\mathbf{a}$ are mitotic (i.e., if A is r.e. with $A \in \mathbf{a}$ than $A = B \cup C$ for some disjoint r.e. sets B and C with $B \equiv_T C \equiv_T A$). Actually we can also get (1.3) from another result we establish using the technique of (1.4) and (1.1):

(1.5) THEOREM. *If* **a** *is array nonrecursive, then* **a** *is not contiguous.*

Note that (1.4) also implies (1.3) via Ladner's result. We shall prove (1.4) and then (1.5) *first* as they require much easier applications of array nonrecursiveness than does (1.1). We hope they will thus illuminate the proof of (1.1).

Notation is standard and follows Soare [17]. We let $A[x] = \{z : z \leq x \ \& \ z \in A\}$. We remain the reader of the convention that all computations, etc., at stage s are bounded by s. We also regard all use functions to be nondecreasing in stage number and argument. For an r.e. set this convention is clear. For a Δ_2 set C with enumeration $C = \lim_s C_s$, it may be that $C_s = C_t$ for some stages $s < t$ yet there is a stage q with $t < q < s$ with $C_s \neq C_q$. The use at stage s must return to the use of stage t. Thus for a Δ_2 set we demand that *new* uses are nondecreasing in stage number and argument. That is, if $\Phi_t(C_t; x)\downarrow$ then $\Phi_t(C_t; z)\downarrow$ for all $z \leq x$, and if $s > t, \Phi_s(C_s; z)\downarrow$ and

$$(\forall q < s)\left[u(\Phi_q(C_q; z) \neq u(\Phi_s(C_s; z)\right]$$

then

$$u(\Phi_t(C_t; z)) \leq u(\Phi_s(C_s; z)).$$

Also $z < u(\Phi_t(C_t; z))$ and if $z_1 < z_2 \leq x$ then

$$u(\Phi_t(C_t; z_1)) < u(\Phi_t(C_t; z_2)).$$

These conventions are quite useful. Finally although the paper is self contained, the reader might find it helpful to read [4] either before, or in conjunction with, the proof of (1.1) in §3.

2. *W*-mitotic and array nonrecursive sets

Following [6] we shall call a strong array $Q = \{D_{g(x)} : x \in \omega\}$ (i.e., where g is recursive, $(\forall x, y)[D_{g(x)} \cap D_{g(x)} = \emptyset]$ and, of course, D_z is the z-th canonical finite set) a *very strong array* if $\cup D_{g(x)} = \omega$ and for all $x < y, |D_{g(x)}| < |D_{g(y)}|$. If Q is a very strong array we shall say a set E is *Q-nonrecursive* if

$$(2.1) \qquad (\forall e)(\exists x)\left[W_e \cap D_{g(x)} = E \cap D_{g(x)}\right].$$

Evidently (2.1) is equivalent to

$$(2.1)' \qquad (\forall e)(\exists^\infty x)(W_e \cap D_{g(x)} = E \cap D_{g(x)}).$$

We call an r.e. degree **a** *array nonrecursive* if there is an r.e. set E of degree **a** and a very strong array Q such that E is Q-nonrecursive. It might appear that the role of Q is crucial but what is important is the "multiple permitting" character as the following basic result from [6] shows.

(2.2) THEOREM [6]. *Let Q be any very strong array and* **a** *array nonrecursive. Then there exists an r.e. set $E \in$* **a** *with E Q-nonrecursive.*

Other results from [6] we will need are summarised in (2.3).

(2.3) THEOREM [6]

(i) *If* **a** *is array nonrecursive and* **b** $>$ **a**, *then* **b** *is array nonrecursive.*

(ii) *There exist array nonrecursive low r.e. degrees.*

(iii) *For all r.e.* **a** $\neq$ **0** *there exists nonzero r.e.* **b** $<$ **a** *with* **b** *array recursive.*

(iv) *If* **a** *is array recursive then* **a** *is* low_2.

We shall now prove (1.4), that is, show array nonrecursive degrees contain non-W-mitotic r.e. sets. Following the notation of Ladner-Sasso [13] we will let "hatted" upper case Greek letters (e.g., $\hat{\Phi}$) denote W-reductions, with the corresponding use function being φ, a partial recursive unary monotone function. Define a set A to be *W-autoreducible* if there is a W-reduction $\hat{\Phi}$ such that, for all $x, \hat{\Phi}(A \cup \{x\}; x) = A(x)$. The intuition is that A can decide x's membership asking only questions involving $y \neq x$. The details of Ladner's corresponding T-degree argument (in [12]) show:

(2.4) THEOREM. *An r.e. set A is W-mitotic iff A is W-auto-reducible.*

Proof. For completeness, we briefly sketch Ladner's argument. More details can be found in [12]. First suppose $A = B \cup C$ is a mitotic splitting of A with $A = \hat{\Phi}(B) = \hat{\Gamma}(C)$. To compute $A(x)$ without using the A-oracle on x perform the computations $\hat{\Phi}(B; x)$ and $\hat{\Gamma}(C; x)$ as usual—where A supplies the correct answers—unless an oracle is questioned about x. If the oracle is questioned about x, automatically answer $x \notin B$ or $x \notin C$ (as appropriate). At least one computation is correct. If they output the same value then this is the correct one. If they output different values then $x \in A$. Note that uses here are $\leq \max\{\varphi(x), \gamma(x)\}$.

On the other hand, if $\hat{\Phi}(A \cup \{x\}; x) = A(x)$ for all x, define

$$l(s) = \max\left\{y : (\forall y \leq x)\left(\hat{\Phi}_s(A_s \cup \{y\}; y) = A_s(y)\right\}.$$

Without loss of generality, as $A \not\equiv_T \emptyset$ we may suppose that we are given enumerations of A and $\hat{\Phi}$ sufficiently fast that

$$(\forall s)[l(s+1) > l(s)] \quad \text{and} \quad (\forall s)(\exists x)(x < l(s) \mathbin{\&} x \in A_{s+1} - A_s).$$

The construction of a W-mitotic splitting $C \cup B$ of A would then run as follows at stage s: enumerate the least $z \in A_{s+1} - A_s$ into $C_{s+1} - C_s$. Enumerate all remaining numbers in $A_{s+1} - A_s$ into $B_{s+1} - B_s$. Note that if $s = s(x) = (\mu s > x)[B_s[\varphi(x)] = B[\varphi(x)]$ then $C_s[x] = C[x]$ since the fact that $\hat{\Phi}$ is an autoreduction ensures that whenever some y enters $A_{s+1} - A_s$, if $y < l(s)$ then some $\hat{y} \neq y$ with $\hat{y} < \varphi(y)$ enters $A_{s+1} - A_s$ too. Hence $C \leq_W B$. The construction evidently gives $B \leq_W C$ by simple permitting.

Thus, to get (1.4) it suffices to show:

(1.4)′ *If* **f** *is an array nonrecursive degree then there exists non-W autoreducible A with* $\deg(A) = \mathbf{f}$.

To achieve (1.4)′ we build A to meet the requirements

$$R_e\colon (\exists x)\big(\hat{\Phi}_e(A \cup \{x\}; x) \neq A(x)\big).$$

Let $Q = \{D_{g(x)}\colon x \in \omega\}$ be a very strong array of the following form: $|D_{g(x)}| > 2^{x+1}$ and

$$(\forall y < x)(\forall p, q)\big[p \in D_{g(y)} \mathbin{\&} q \in D_{g(x)} \to p < q\big].$$

Let F be an r.e. Q-nonrecursive set of degree **f** and suppose $F = f(\omega)$ with f 1-1 recursive.

At each stage s we let $\{a_{i,s}\colon i \in \omega\}$ list $\overline{A}_s \cap \omega^{(0)}$ in order. Here $\omega^{(j)} = \{\langle j, x\rangle\colon x \in \omega\}$. For the sake of this construction it is convenient to regard $\langle\ ,\ \rangle$ as monotone in the second variable, monotone in both variables for $\langle x, y\rangle$ with $x \geq 1$, and enumerated so that the following properties hold (this needs two different pairings and is definitely not obtained by repeated use of the pairing function):

$$(2.5) \qquad (\forall e)(\forall z \geq e)\big[|\{y\colon \langle 0, y\rangle < \langle e+1, \langle z, z\rangle\rangle\}|\, |z\big],$$

and

$$(2.6) \qquad (\forall e)(\forall z \geq e)(\forall m \leq z)\big[\{y\colon \langle 0, y\rangle < \langle e+1, \langle m, z\rangle\rangle\} = \{y\colon \langle 0, y\rangle < \langle e+1, \langle z, z\rangle\rangle\}\big]$$

Henceforth we will write $\langle x, y, z\rangle$ for $\langle x, \langle y, z\rangle\rangle$. To achieve $A \equiv_T F$ we employ coding markers $\wedge(i, s)$ which rest on (some) members of $\overline{A}_s \cap \omega^{(0)}$.

Initially place $\{\wedge(j,0)\colon j \in \omega\}$ on elements of $\omega^{(0)}$ subject to the rules

(i) $\wedge(j,0) < \wedge(j+1,0)$,
(ii) $\wedge(j,0) \in D_{g(x)}$ and $\wedge(k,0) \in D_{g(x)}$ implies $j = k$,
(iii) $\wedge(j,0) \in D_{g(x)}$ implies $x < j$.

We ensure that

(i) $(\forall i < j)[\wedge(i,s) < \wedge(j,s)]$ and $(\forall s)[\wedge(i,s+1) \geq \wedge(i,s)]$:
(ii) $\wedge(i,s+1) \neq \wedge(i,s)$ implies $F_{s+1}[m(i)] \neq F_s[m(i)]$ where $m(i) = \max\{x\colon x \in D_{g(i)}\}$;
(iii) $F_{s+1}[j] \neq F_s[j]$ implies that $A_{s+1}[\wedge(j,s)] \neq A_s[\wedge(j,s)]$;
(iv) $\wedge(j,s+1) \neq \wedge(j,s)$ implies $y \in A_{s+1} - A_s$ for some $y \leq \wedge(j,s)$;
(v) $A_{s+1} - A_s = \{y\}$ for some y;
(vi) $A_{s+1}[i] \neq A_s[i]$ implies that $F_{s+1}[m(\langle i,i,i\rangle)] \neq F_s[m(\langle i,i,i\rangle)]$;
(vii) (consequence of (ii)) $\lim_s \wedge(i,s) = \wedge(i)$ exists;
(viii) $(\forall s,j)(\exists k)(k \geq j$ and $\wedge(j,s+1) = \wedge(k,s))$.

It is routine to verify that if our construction obeys (i)–(vii) then $A \equiv_T F$ as follows. First $F \leq_T A$. Given z, to compute $F(z)$, find the least stage s where $A_s[(\wedge(z,s)] = A[\wedge(z,s)]$. Then $F_s(z) = F(z)$ since if $F_s[z\} \neq F[z]$ at the first stage $t \geq s$ where $F_{t+1}[z] \neq F_s[z]$, (iii) would cause

$$A_t[\wedge(z,s)] \neq A_{t+1}[\wedge(z,s)].$$

Also $A \leq_T F$ since to compute $A[z]$ find tne least stage s where

$$F_s[m(\langle z,z,z\rangle)] = F[m(\langle z,z,z\rangle)].$$

Then $A_s[z] = A[z]$.

To satisfy the R_e above, our idea is to try to build an auxiliary r.e. set $H_e = \bigcup_s H_{e,s}$ where $H_e \subseteq \omega^{(e+1)}$ which we use to force F to give many permissions. We will argue that if F fails to give enough permissions to meet R_e, then in fact H_e will be a witness to the Q-recursiveness of F, a contradiction.

Since our argument is finite injury allowing us to initialise H_e—when R_j for $j < e$ requires attention—it will suffice to describe the strategy for a single R_e. Let

$$l(e,s) = \max\left\{x\colon (\forall y < x)\left(\hat{\Phi}_{e,s}(A_s \cup \{y\}\colon y) = A_s(y)\right)\right\}.$$

The construction runs as follows. Unless otherwise directed (by an R_j) always enumerate $\wedge(f(s),s)$ into $A_{s+1} - A_s$, and set $\wedge(j,s+1) = a_{j+\wedge(f(s),s)+s,s+1}$ for all $j \in \omega$ with $j \geq f(s)$ (we call this kicking $\wedge(j,s+1)$).

Note that this keeps $\wedge(j, s+1) \in \omega^{(0)}$ inductively. We now need do nothing for the sake of R_e until we see a stage s occur and a number $z(0) = \langle e + 1, y, y \rangle$ for some $y > e$.

$$(2.7) \qquad \begin{aligned} &(\exists t \leq s)[l(e,t) > z(0)], \\ &z(0) > \wedge(z,s) \text{ for some } z \text{ with } z = f(s). \end{aligned}$$

It is not difficult to see that if a stage where (2.7) occurs does not eventuate and R_e fails to be met then by a permitting argument F would be recursive. To compute $F(z)$ find the least stage s such that for some $y > e$ we have $l(e,s) > \langle e+1, y, y\rangle > \langle e+1, 0, y\rangle > \wedge(z,s)$. Then as (2.7) fails it must be that $F_s[z] = F[z]$.

Before we begin the discussion of the cycle for $z(0)$, it is worthwhile to analyse the key elements of the construction to follow and, in particular, the use of array nonrecursiveness. First, if it were unnecessary to keep $A \leq_T F$ but simply code F into A, one way to meet R_e (and so simply build a non-W-mitotic r.e. set $\geq_T F$) would be to await a stage s where

$$l(e,s) > \langle e+1, y, y\rangle > \wedge(z,s)$$

and proceed in two steps. First we would enumerate $\wedge(z,s)$ into $A_{s+1} - A_s$ and then kick the markers $\wedge(z',s)$ for $z' \geq z$. Again we remark that as the argument is finite injury, we could initialize the R_n for $n > e$ and ask them to only thereafter use $\wedge(k,t)$ for $k > \max\{z, n\}$.

To then temporarily win R_e, we would await a stage $t > s$ such that $l(e,t) > z(0)$. If such a stage does not occur then $l(e,s) \not\to \infty$. Consider the situation at such a stage t. Certainly we could create a temporary disagreement by enumerating $\langle e+1, 0, y\rangle = \hat{y}$ into $A_{t+1} - A_t$ which would make

$$\hat{\Phi}_{e,t+1}(A_{t+1} \cup \{\hat{y}\}, \hat{y}) = \Phi_{e,t}(A_t \cup \{\hat{y}\}, \hat{y}) = A_t(\hat{y}) \neq A_{t+1}(\hat{y})$$

The other thing to note is that since we are using W-reductions the use functions are unchanged: $\varphi_e(A_t \cup \{\hat{z}(0)\}; \hat{z}(0)) = \varphi_e(A_s \cup \{\hat{z}(0)\}; \hat{z}(0)) = \varphi$, say. In particular, as we kicked $\wedge(k,s)$ above s for all $k \geq z$, it follows that $\wedge(k,s) > \varphi$ for all $k \geq z$.

The relevance of this is the following. For our disagreement at $z(0)$ to be injured, injury can only occur due to higher priority activity, or *due to the coding* of $\wedge(q,s) = \wedge(q,t)$ *for some* $q < z$, since inductively $(\forall t > s)(\forall k > z)(\wedge(k,t) \geq \wedge(k,s+1) > s > \varphi)$.

In the usual finite injury way, we pretend R_e is the highest priority unsatisfied requirement around, and then the only injuries can be caused by coding.

Thus suppose at some stage $g > t$ we see $\wedge(r,s) = \wedge(r,t) < \varphi$ enter A upsetting the disagreement at $\langle e+1, 0, y\rangle = \hat{y}$. We can then begin the "cycle" for $z(0)$ anew and await a stage q where $l(e,q) > z(0)$ again. We then win temporarily by enumerating $\langle e+1, 1, y\rangle$ into A creating a disagreement at $\langle e+1, 1, y\rangle$. Note that by kicking, for all $\hat{r} \geq r$, $\wedge(\hat{r}, q) \geq \wedge(\hat{r}, g) > g > \varphi$. In subsequent cycles we would use $\langle e+1, 2, y\rangle$ etc. The reader should note that as $\wedge(j,s) \in \omega^{(0)}$, (2.5) and (2.6) together with our kicking strategy will ensure that we would meet R_e after at most z cycles and so only "use up" $\langle e+1, 0, y\rangle, \ldots, \langle e+1, z, y\rangle$, all of whom are below $\langle e+1, y, y\rangle$ by (2.6). Thus after at most $j < z$ attacks we would succeed in meeting R_e.

The readers should keep the above in the back of their minds when we do the full construction as a sort of "inner strategy". Our big problem with this strategy is that we must also obey permitting rules. We are only allowed to enumerate into A when F says we can. Thus the first problem we would need to overcome is to force F to permit us to enumerate many numbers like $\langle e+1, 0, y\rangle, \ldots, \langle e+1, j, y\rangle$ in the above. Indeed the problem is compounded in that if F permits i then by (iii) of our marker rules we are forced to code some $y \leq \wedge(i,s)$ into A. This causes a severe problem since such coding might interfere with any potential disagreement we are attempting to cause at $\langle e+1, i, y\rangle$.

By initalization, we know that initially $H_{e,s} = \emptyset$. The array nonrecursiveness of F ensures that for some x, $D_{g(x)} \cap F = H_e \cap D_{g(x)}$. It is within *our* power to enumerate numbers into H_e. We must arrange things so that if R_e fails then $(aax)(D_{g(x)} \cap F \neq H_e \cap D_{g(x)})$. It is very important that the reader note that if ever we see a stage where $n \in D_{g(x)} \cap F_s$ for some $n \notin H_{e,s}$ we can ensure $D_{g(x)} \cap F \neq H_e \cap F$ by simply being "conservative" on H_e and asking

$$H_e \cap D_{g(x)} = H_{e,s} \cap D_{g(x)}.$$

Thus the only $D_{g(x)}$ for which we really need to enumerate numbers into H_e, for the sake of R_e, are those with $H_{e,s} \cap D_{g(x)} \supseteq F_s \cap D_{g(x)}$ for all s.

The basic idea is that when we wish to enumerate $\langle e+1, i, y\rangle$ into A for the sake of R_e, we shall cause $F_s \cap D_{g(z)} \neq H_{e,s} \cap D_{g(z)}$ as conservatively as possible (i.e., no enumeration, or enumerating *one* number into $H_{e,s+1} - H_{e,s}$). Now either F can respond by correcting $F_t \cap D_{g(z)} = H_{e,s} \cap D_{g(z)}$ at some $t > s$ giving us our desired permission or we have made sure that $F \cap D_{g(z)} \neq H_e \cap D_{g(z)}$. Note that if $F \cap D_{g(z)} = H_e \cap D_{g(z)}$ then we can win R_e on one $\langle e+1, i, y\rangle$ since there are at most $z < y$ coding injuries (as we noted earlier), yet it is within our power to cause $|D_{g(z)}| > z$ permissions in F.

Unfortunately, we do not know if $D_{g(z)} \cap F = H_e \cap D_{g(z)}$ for this z. The idea is that while it appears that $D_{g(z)} \cap F \neq H_e \cap D_{g(z)}$ we begin an inductive strategy on subsequent $z' > z$. Matters are arranged so that if R_e fails then eventually almost all of the $D_{g(x)}$'s are devoted to solving R_e and so giving the failure of the array nonrecursiveness of F.

We now give the details of the above.

The cycle for $z(0)$.

1. When 2.7 occurs, enumerate $\wedge(z, s)$ into A_{s+1} kicking $\wedge(j, s+1)$ for all $j \geq z$. Also restrain $A[s]$ with priority e from access to any R_k for $k > e$ by initialization.

2. We wait for the least stage $u \geq s+1$ such that $l(e, u) > \max\{l(e, t): t < u\} > z(0)$. Note if u fails to occur then $l(e, s) \not\to \infty$. We remind the reader that if $\wedge(j, t) \in A_{t+1} - A_t$ for some $s < t < u$ then $\wedge(j, t)$ is kicked (if it is $< \langle e+1, y, y\rangle$). Note that once stage u occurs, a typical situation will be as given in Diagram 1 below.

$$\underset{|}{\wedge(0, u+1)} \quad \underset{|}{\wedge(1, u+1)} \quad \underset{|}{\wedge(z-1, u+1)} \quad \underset{|}{\varphi_e(\langle e+1, 0, y\rangle)} \quad \underset{|}{\varphi_e(z(0))} \quad \underset{|}{\wedge(z, u+1)} \quad \underset{|}{\wedge(z+1, u+1)} \quad A_{u+1}$$

DIAGRAM 1

3. When we see $l(e, u) > z(0)$ then for each $q \leq z(0)$ with $q \geq e$ make

$$D_{g(q)} \cap H_{e, u+1} \neq D_{g(q)} \cap F_u$$

as conservatively as possible. That is if $D_{g(q)} \cap H_{e,u} \neq D_{g(q)} \cap F_u$ already do nothing, and if $D_{g(q)} \cap H_{e,u} = D_{g(q)} \cap F_u$ enumerate the least $z \notin H_{e,u}$ with $z \in D_{g(q)}$ into $H_{e,u+1}$. (As above, such a z will exist.)

4. Wait till we see a least stage $q \geq u+1$ such that one of the following holds.

4(*a*). $F_{q+1}[m(z(0))] \neq F_q[m(z(0))]$, but for all i with $\wedge(i, u) < z(0)$, $F_{q+1}[i] = F_q[i]$.

Action. Set $A_{q+1} = A_q \cup \{\langle e+1, 0, y\rangle\}$. As we discussed in the inner strategy, this will create a temporary disagreement at $\langle e+1, 0, y\rangle$ and will complete the cycle for $z(0)$ unless some R_k of higher priority acts, or coding affects matters via step 5 below.

4(*b*). $F_{q+1}[i] \neq F_q[i]$ for some $\wedge(i, q) = \wedge(i, s) < z(0)$.

Action. Enumerate $\wedge(f(q), q)$ into $A_{q+1} - A_q$ and kick as usual. If $f(q) \leq e$ we initialise and so we pretend to be in stages where $F_s[e] = F[e]$. We go back to step 2 in the cycle for $z(0)$. While we wait for 2 to occur if 4(*b*) pertains to some $f(r) < i$ continue to kick $\wedge(f(r), r)$.

Analysis of this outcome. It is very important to note the effect of outcome 4(*b*). First the reader should note that for all p with $e \leq p < j$ where

$i \in D_{g(j)}$ we have (still)

$$D_{g(p)} \cap H_{e,q} \neq D_{g(p)} \cap F_q.$$

Furthermore the reader should note that if $i \in D_{g(j)}$ and $i \neq \mu z(z \in D_{g(j)})$, then by our action in step 3—of picking the *least* available member of $D_{g(j)}$—we also know that

$$D_{g(j)} \cap H_{e,q} \neq D_{g(j)} \cap F_q.$$

Indeed, by our conservative strategy since $i \in D_{g(j)} \cap F_q$ and $i \notin H_{e,q}$ we actually know we can force $D_{g(j)} \cap H_e \neq D_{g(j)} \cap F_q = D_{g(j)} \cap F$.

The point of all of the above is this. Consider the next time 3 pertains to $z(0)$ say at stage n. Let r be the least number to have occurred in $F_n - F_q$ and suppose $r \in D_{g(p)}$. Then we know that for all k with $e \leq k < p$ we still have

$$D_{g(k)} \cap H_{e,n} \neq D_{g(k)} \cap F_n.$$

Furthermore, for all m with $g(p) \leq g(m) \leq z(0)$ it is possible for $D_{g(m)} \cap H_{en,n} = D_{g(m)} \cap F_n$ only if exactly the least member of $D_{g(m)} \cap H_{e,q+1}) - (D_{g(m)} \cap H_{e,q})$ entered $F \cap D_{g(m)}$. In particular for an m for which $D_{g(m)} \cap H_{e,n} = D_{g(m)} \cap F_n$ we know that, by kicking, $\wedge(i,n) > q$ for all $i \in D_{g(m)}$. In other words, if F now permits $m(m)$ via some $i \in D_{g(m)} \cap F_n$ *then this permission is helpful in the sense that* $4(a)$ *pertains to* $z(0)$. Thus, when 3 again pertains, we will cause

$$D_{g(m)} \cap H_{e,n+1} \neq D_{g(m)} \cap F_n \quad \text{for all } g(e) \leq g(m) \leq z(0).$$

This will cause *no enumeration* for all $D_{g(k)}$ with $g(e) \leq g(k) < g(p)$ *and at most one enumeration for all* $D_{g(k)}$ with $g(p) \leq g(k) \leq z(0)$. Moreover, by (2.5) and (2.6), $4(b)$ can only now pertain $< g(p) - 1$ more times (as all the other markers are kicked). It therefore follows that for all m with $g(e) \leq g(m) \leq z(0)$, it is always possible to cause $D_{g(m)} \cap H_{e,n+1} \neq D_{g(m)} \cap F_n$ as $D_{g(m)}$ has 2^{m+1} members.

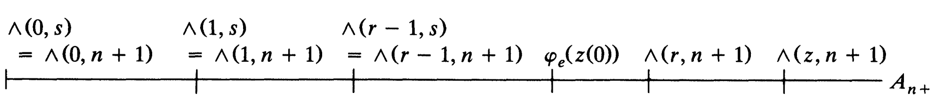

DIAGRAM 2

(The only unhelpful permission can occur due to $\wedge(j,s) = \wedge(j,n+1)$ for $j \leq r - 1$ after stage n.)

5. If case 4(a) pertains then we obviously have a disagreement that wins unless this is injured by F-coding. Again this means that some i enters $F_t - F_q$ with $\wedge(i,t) = \wedge(i,s) < z(0)$. In this case we must begin our cycle anew.

The only problem is that we have used up $\langle e+1, 0, y\rangle$. The only difference now for $z(0)$ is that if 4(a) again pertains we now enumerate $\langle e+1, 1, y\rangle$ (and in subsequent cycles $\langle e+1, i, y\rangle\rangle$ in order), exactly as in the inner strategy. For the same reasons as in 4(b) there can be at most $< y$ injuries.

6. If R_e is to fail, then, the only possibility is that we never get to a permanent disagreement in 4(a). This will mean that for *all* $\hat{y} \geq e$ and $\hat{y} \leq z(0)$ in the limit we will have $F \cap D_{g(\hat{y})} \neq H_e \cap D_{g(\hat{y})}$. The obvious strategy is the inductive one. Whilst we are in a cycle for $z(0)$ after we create $F_u \cap D_{g(\hat{y})} \neq H_{e,u} \cap D_{g(\hat{y})}$ disagreements for all such $\hat{y}$, we wait to try to find a new $z(1) > u$ to begin a cycle for, assuming $z(0)$ is no longer active.

Specifically we proceed as follows. Once stage 3 occurs and we are waiting for 4 to occur we believe that $z(0)$ is permanently stuck we therefore search for the least $z(1) > z(0)$ with $z(1) = \langle e+1, m, m\rangle$ for some m such that (2.7) applies to $z(1)$ (in place of $z(0)$).

The idea is to then begin a cycle for $z(1)$, and continue as we did for $z(0)$ until we get to perform 4 on $z(1)$, or we get stuck waiting for 4 on $z(1)$, or find that the belief that we were stuck on $z(0)$ was wrong. The reader should note that $z(0)$ is of higher priority than $z(1)$ and so even though we appear to have met R_e via $z(1)$ (i.e. via 4(a)) should we get a chance to make R_e temporarily satisfied via $z(0)$ we will do so (as this attack is more likely to succeed).

A typical situation is given in Diagram 3 below.

$$\wedge(0,v) \quad \wedge(1,v) \quad \wedge(z',v) \quad \varphi_e(z(0)) \quad \wedge(z'+1,v) \quad \wedge(z'',v) \quad \varphi_e(z(1)) \quad \wedge(z''+1,v) \qquad A_v$$

DIAGRAM 3

Here $z' < y$ where $z(0) = \langle e+1, y, y\rangle$, and $z'' < \hat{y}$ where $z(1) = \langle e + 1, \hat{y}, \hat{y}\rangle$.

Note that in the situation of Diagram 3, the only *injuries* to $z(0)$'s set up can come from F coding r for $r < z'$ and the only injuries to $z(1)$'s set up, that do not injure $z(0)$'s set up can come from r with $z' \leq r < z''$. Inductively we will know that for all x with $e \leq x \leq z(0)$,

$$D_{g(x)} \cap F_v \neq D_{g(x)} \cap H_{e,v}.$$

The important point is that for any m, we will need to enumerate a new element into D_m to make

$$D_{g(x)} \cap F_v \neq D_{g(x)} \cap H_{e,s}$$

only if F causes a coding that is not helpful to requirement R_e, *and furthermore* a coding for r with $\wedge(r, v) \in D_{g(x)}$. By the movement of the markers, any future *unhelpful* coding must involve $D_{g(y)}$ for $y < x$ (indeed $y < r$). (If the reader does not see this he or she is advised to go back to the paragraph following (2.6).)

The same conditions hold for $z(1)$ and so the same counting holds. It is clear that this idea extends to $n > 1$ $z(i)$'s. Furthermore we will appoint infinitely many $z(i)$'s only if $\limsup l(e, s) \to \infty$ and since we are dealing with W-reductions, this occurs only if

$$(\forall x)\left[\hat{\Phi}_e(A \cup \{x\}; x) = A(x)\right].$$

By the counting argument described above, should R_e fail to be met then

$$(\forall m \geq e) D_{g(m)} \cap H_e \neq D_{g(m)} \cap F,$$

a contradiction. It follows that R_e can receive attention at most finitely often and hence a standard application of the finite injury method gives the result. ∎

As we mentioned in the introduction, the ideas above can also be used to show:

(2.8) THEOREM. *Suppose* **f** *is array nonrecursive. Then* **f** *is not contiguous.*

Proof (Sketch). We only sketch the proof due to its similarities with the previous result. Suppose **f** is array nonrecursive and F is Q-nonrecursive, etc., as in the previous result. We construct $A = \bigcup_s A_s$ with $A \equiv_T F$ and an r.e. set $B \leq_W F$ such that we meet for all e,

$$R_e: \hat{\Phi}_e(A) \neq B.$$

Again we use marker coding via $\wedge(i, s)$ and auxiliary r.e. sets $H_e = \bigcup_s H_{e,s}$ to force permissions. The reduction $B \leq_W F$ is given via $x \in B$ iff $x \in B_s$ where $s = \mu t(F_t[m(x)] = F[m(x)])$. We let $\{a_{i,s}: i \in \omega\}$ list $\overline{A}_s$ in order.

The construction of A.
Stage 0. Set $A_0 = \emptyset$ and $\wedge(i, 0) = a_{i,0} = i$.
Stage $s + 1$. Set

$$A_{s+1} = A_s \cup \{\wedge(f(s), s)\}.$$

Kick the markers $\wedge(j, s+1)$ for $j \geq f(s)$ (e.g., set $\wedge(j, s+1) = a_{j+\wedge(j,s)+s+1, s+1}$).
Set $A = \bigcup_s A_s$.

End of construction

We refer to A so constructed as the *kick set of* F, and write $A = K(F)$. It will also be used in §3.

We show that R_e is automatically met by such A for suitable B. As in (1.4)′ we devote $\omega^{(e+1)}$ to R_e and shall assume the pairing of (2.5) and (2.6). The attack on R_e runs as follows. Let

$$l(e, s) = \max\left\{x : (\forall y < x)\left[\hat{\Phi}_{e,s}(A_s; x) = B_s(y)\right]\right\}.$$

The cycle for $x \geq e$

1. Wait till F permits x at s and $l(e, t) \geq \langle e+1, \langle x, x\rangle\rangle$ for some $t \leq s$. (Now x becomes "activated" as in (1.4)′.)

2. At the least t where $l(e, t) > \langle e+1, \langle x, s\rangle\rangle$ and $t \geq s$, as in (1.4)′ for each $y \geq e$ and $y \leq x$ make $D_{g(y)} \cap F_t \neq H_{e,t+1} \cap D_{g(y)}$.

3. Whilst $l(e, u) > \langle e+1, \langle x, x\rangle\rangle$, if we see a stage $u > t$ where F permits $m(x)$ and $F_u[x] = F_t[x]$, enumerate $\langle e+1, \langle 0, x\rangle\rangle$ into $B_{u+1} - B_u$. As we see below, this temporarily satisfies R_e, and uses up $\langle e+1, \langle 0, x\rangle\rangle$ (but leaves $\langle e+1, \langle i, x\rangle\rangle$ for $0 < i \leq x$ for subsequent attacks). This corresponds to outcomes 4(a) of (1.4)′. The reason it satisfies R_e is that since F permitted x in step 1 we know that for all $y \geq x$, $\wedge(y, t) > s > \varphi_e(A; \langle e+1, \langle x, x\rangle\rangle)$—since we are dealing with W-reductions. Thus at stage u we know $A_{u+1}[s] = A_u[s]$ but

$$B_{u+1}(\langle e+1, \langle 0, x\rangle\rangle) \neq B_u(\langle e+1, \langle 0, x\rangle\rangle)$$

causing a disagreement at $\langle e+1, \langle 0, x\rangle\rangle$.

4. If we see a stage $w > t$ where F permits x then we return to step 2 as in (1.4)′. This is true whether or not step 3 occurs. There can only be x such injuries by kicking so that for some $i \leq x$ we get a disagreement on $\langle e+1, \langle i, x\rangle\rangle$. Also only z members of $D_{g(z)}$ can be used.

It is (hopefully) clear that the above strategy meets R_e for the same reasons as (1.4)′ and the argument will be finite (bounded) injury.

The crucial point of the kicking procedure is that whenever there is an injurious coding of some $\wedge(i, s)$ in the sense of 4(b) of (1.4)′ or step 4 of the above, this kicks all the markers $\wedge(j, s)$ for $j \geq i$ into noninjurious positions, and furthermore ensures that thereafter we have at most i possible further injurious codings. The other point is that it leaves the disagreements of

$H_{e,s} \cap D_{g(m)} \neq F_u \cap D_{g(m)}$ for those $m < k$—where $i \in D_{g(k)}$—alone. This means that we can't use up all of a $D_{g(m)}$, as there can only be $< |D_{g(m)}|$ injuries. ∎

The argument of the next section uses the ideas of the above, but is more difficult since we are no longer dealing with W-reductions. We will nevertheless use a kicking strategy and use the fact that we are building two sets to control the sorts of codings injuries we can experience, and hence ensure that all of a $D_{g(m)}$ cannot be used up. Roughly speaking the point with W-reductions is that after kicking we can clear the relevant use (as it can't change). In the next section, we will be allowed to clear one *use* by enumerating in the *other* set since we will be constructing two sets but only need to code into one.

3. The diamond theorem

Let F, Q, etc. be as in §2. Let $A = K(F)$ as in (2.8). We shall also construct d.r.e. sets B and C with $B \oplus C \equiv_T A$ by Δ_2 permitting. The requirements we must satisfy are

$$P_{2e}\colon \overline{B} \neq W_e,\ P_{2e+1}\colon \overline{C} \neq W_e$$
$$N_e\colon \Phi_e(B) = \Phi_e(C) = f \text{ total} \Rightarrow f \text{ recursive}.$$

We remark that $A = K(F)$ is really not necessary to this construction, but we hope that it will help the reader visualize the reductions.

To ensure that $B \oplus C \geq_T F$ we will ensure that if $\wedge(i, s+1) \neq \wedge(i, s)$ then either

$$B[\wedge(i,s)] \neq B_s[\wedge(i,s)) \quad \text{or} \quad C[\wedge(i,s] \neq C_s[\wedge(i,s)].$$

We then get $B \oplus C \geq_T F$ since if t is a stage where $B_t[\wedge(i,t)] = B[\wedge(i,t)]$ and $C_t[\wedge(i,t)] = C[\wedge(i,t)]$ then $F[i] = F_t[i]$. To ensure $B \oplus C \leq_T F$ we only allow $B_s[\wedge(i,s)] \neq B[\wedge(i,s)]$ if $F_s[m(i)] \neq F[m(i)]$ and similarly for C ($m(i) = \max\{x\colon x \in D_{g(i)}\}$ as in §2). Although we are using Δ_2 permitting, the reader should note that the construction will additionally obey the following rule:

If $F[i] \neq F_t[i]$ then for some $j \leq i$, $\wedge(j,t) \in C - C_t$ or $\wedge(j,t) \in B - B_t$. That is we will cause some change by *enumeration*. This formulation makes the Δ_2 coding much easier to visualize. We call this property (3.1).

The principal difference between our construction here and that of §2 is that although this one is a finite injury argument, it has a hidden π_2 outcome. That is, the key to using the finitary techniques of §2 will be that requirements *receive attention* and thus do something *active* only finitely often, the "π_2-outcome" will be for R_e to be *monitored* infinitely often. Monitoring causes no *enumeration* or *extraction*. This distinction is crucial to the success of the construction.

The P_e are met by a usual Friedberg strategy where we ensure that eventually if W_e looks like $\overline{B}$ (or $\overline{C}$) then some $\wedge(i, s)$ will be forced into $W_{e,s} \cap B_s$ and protected from *extraction* with priority e. As the restraints of N_j for $j \leq e$ are finite this meets the P_e as usual. We shall meet the N_e by a strategy somewhat along the lines of [4], but with some modifications. Other 'diamond' strategies (e.g., Cooper [3]) seem inappropriate.

We need some auxiliary functions

$$u(e, x, s) = u(\Phi_{e,s}(B_s; x)),$$
$$\hat{u}(e, x, s) = u(\Phi_{e,s}(C_s; x))$$

When the meaning is clear from context we will write $u(x, s)$ or even $u(s)$ to save on notation.

$$m(e, x, s) = \max\{u(e, x, s), \hat{u}(e, x, s)\},$$
$$\hat{m}(e, x, s) = \min\{u(e, x, s), \hat{u}(e, x, s)\}.$$

The basic module

One of the distinctions between this paper and [4] is that if N_e's outcome is infinitary it may still be possible for it to have finite effect (restraint) on the construction due to *failure of permission* (more on this later).

When some y occurs in $F_{s+1} - F_s$ we shall add $\wedge(y, s)$ into one, or both, of B_s or C_s as determined by the other requirements (e.g. the other N_j or the Friedberg requirements P_j). Should no requirement make any specific request we add the $\wedge(y, s)$ to both B_s and C_s. Let

$$l(e, s) = \max\{x: (\forall y < x)(\Phi_{e,s}(B_s; y) = \Phi_{e,s}(C_s; y))\}$$

and

$$ml(e, s) = \max\{l(e, t): t < s\}.$$

We say s is *e-expansionary* if $l(e, s) > ml(e, s)$. Set

$$ls(e, s) = \max\{0, t: t < s \ \& \ t \text{ is } e\text{-expansionary}\}.$$

We shall be monitoring $l(e, s)$. Of course if $(\exists s_0)(\forall s > s_0)[l(e, s) \leq ml(e, s_0)]$ then N_e only has finite effect and is met by divergence. We call this outcome d. Assuming this fails to pertain, we search for a disagreement. Thus let $s(1)$ and $s(2)$ be e-expansionary stages with $s(1) = ls(e, s(2))$. At stage $s(2)$ we see how our previously seen computations are going. If for all $z \leq ml(e, s(1))$ we have

$$\Phi_{e,s(1)}(B_{s(1)}; z) = \Phi_{e,s(2)}(B_{s(2)}; z) = \Phi_{e,s(2)}(C_{s(2)}; z) = \Phi_{e,s(1)}(C_{s(1)}; z)$$

then N_e need do nothing. If this outcome is the only outcome almost always, then N_e's effect is finite (again), and to compute $\Phi_e(B; z)$ find the least e-expansionary s with $l(e, s) > z$ for some $s > s_0$ for some parameter s_0 (assuming $\Phi_e(B) = \Phi_e(C)$).

The dangerous case occurs if we see some $z(1) = z < l(e, s(1))$ with

$$\Phi_{e,s(1)}(B_{s(1)}; z) \neq \Phi_{e,s(2)}(B_{s(2)}; z), \text{ and so} \tag{3.2}$$

$$\Phi_{e,s(1)}(C_{s(1)}; z) \neq \Phi_{e,s(2)}(C_{s(2)}; z)$$

What we would *like* to do is use (3.2) to create, and preserve, a disagreement at $z(1)$. One obvious way to do this is to reset $C_{s(2)+1}[s(1)] = C_{s(1)}[s(1)]$ which would cause a disagreement. However, we must do this—or something like it —in such a way as to still code F in: perhaps we were using $C_{s(2)}[s(2)]$ to code the fact that some j has occurred in $F_{s(2)} - F_{s(1)}$. Furthermore we will need F to permit any C-change or B-change. What we shall do is first get an "even smaller" number coded into "both sides" of B and C.

Let b be the least number that has entered or been extracted from B or C between stages $s(2)$ and $s(1)$. Let $\hat{y}(1)$ be largest with $\wedge(\hat{y}(1), s(1)) < b$. Note that $\wedge(\hat{y}(1), s(2)) = \wedge(\hat{y}(1), s(1))$ by property (3.1).

The cycle for $z(1)$.

1. For all $x \geq e$ and $x \leq \hat{y}(1, s(2))$, make $D_{g(x)} \cap H_{s(2)+1} \neq F_{s(2)} \cap D_{g(x)}$. Do this as in §2, conservatively. N_e restrains B and C through $s(2)$.

2. Wait till we see an F-permission on $m(\hat{y}(1))$. If F permits $m(\hat{y}(1))$ at stage $n(1) \geq s(1)$, for $k \in F_{n(1)} - F_{n(1)-1}$ enumerate $\wedge(k, n(1)) = \wedge(k, s(1))$ into both $C_{n(1)} - C_{n(1)-1}$ and $B_{n(1)} - B_{n(1)-1}$ and now wait for the $\hat{\Phi}_e$ computations to recover. Whilst we are waiting N_e keeps its restraints at $s(2)$ and should some $\hat{k} < k$ occur in $F_p - F_{n(1)}$ we put $\wedge(\hat{k}, p) = \wedge(\hat{k}, g(1))$ into both B and C.

3. Wait till we see a stage $s(3) > s(2)$ such that $l(e, s(3)) > ml(e, s(3))$. When this occurs raise N_e's restraints to $s(3)$. Of course should $s(3)$ not occur we meet N_e by divergence. Let $y(1) = y(1, s(2))$ be the least number to occur in $(B_{s(3)} \cup C_{s(3)}) - (B_{s(1)} \cup C_{s(1)})$. Note that $\wedge(y(1), s(2)) = \wedge(y(1), s(1))$. Now as $s(3)$ is e-expansionary, we shall know that one of (3.3) or (3.4) below

pertains.

$$\Phi_{e,s(3)}(B_{s(3)};z) = \Phi_{e,s(1)}(B_{s(1)};z), \quad \text{and so} \tag{3.3}$$

$$\Phi_{e,s(3)}(C_{s(3)};z) = \Phi_{e,s(1)}(C_{s(1)};z)$$

$$\Phi_{e,s(3)}(B_{s(3)};z) = \Phi_{e,s(2)}(B_{s(2)};z), \quad \text{and so} \tag{3.4}$$

$$\Phi_{e,s(3)}(C_{s(3)};z) = \Phi_{e,s(2)}(C_{s(2)};z)$$

Now we can use whichever of (3.3) or (3.4) pertains to create a disagreement (provided we get the relevant permission). If (3.3) pertains then what we would like to do is set

$$C_{s(3)+1}[s(2)] = C_{s(2)}[s(2)]$$

but keep

$$B_{s(3)+1}[s(3)] = B_{s(3)}[s(3)].$$

This would create a disagreement at z by (3.2) and (3.4). The relevant permission needed is that we need F to permit $m(y(1))$ but *not* permit $y(1)$. For then if F permits $h > y(1)$ with $h \leq m(y(1))$ we can put $\wedge(h, s(3))$ into C for coding F into $B \cup C$ but reset C back to $C_{s(2)}[s(2)]$. The point is that by kicking $\wedge(h, s(3)) > s(2)$ as $\wedge(y(1), s(2))$ was enumerated *before* $s(3)$.

Similarly if (3.4) pertains, we'd like to reset $C_{s(3)+1}[s(1)]$ back to $C_{s(1)}[s(1)]$. As these situations are dual, for simplicity we'll suppse (3.4) pertains, and continue the cycle below.

4. For all $x \geq e$, $x \leq y(1, s(2))$, make $D_{g(x)} \cap H_{s(2)+1} \neq F_{s(2)} \cap D_{g(x)}$ as in 1.

5. Wait until we see an F-permission.

Case 1. F permits $m(y(1))$ at stage $t = t(1) \geq s(1)$ but $F_{t(1)}[y(1)] = F_{s(3)}[y(1)]$.

Action. For $h \in F_t - F_{t-1}$, enumerate $\wedge(h, t) = \wedge(h, s(3))$ into $C_t - C_{t-1}$ and otherwise set $C_t[s(1)] = C_{s(1)}[s(1)]$. Note $\wedge(h, s(2)) > s(1)$, and so this is possible and gives a disagreement at $z(1)$.

Case 2. F permits $y(1)$ at some $t(1) \geq s(3)$.

Action. If $g \in F_{t(1)} - F_{t(1)-1}$, enumerate $\wedge(g, t(1)) = \wedge(g, s(1))$ into *both* $B_{t(1)}$ and $C_{t(1)}$. Note that if $\hat{g}$ is the least such g to enter F then we can again go to 3 in the cycle for $z(1)$ and we can use $\hat{g} = y(1, t(2))$ in place of $y(1, s(1))$. Again, as in §2 this should be regarded as *injury* but *this* sort of injury can only occur x times to $D_{g(x)}$ (for any x) for the same reasons as §2. Note that if this case pertains and we again return to stage 4, our strategy of setting $C[s(1)] = C_{s(1)}[s(1)]$ or $C[s(2)] = C_{s(2)}[s(2)]$ will be the same. This is also true after 6. However, in this construction another form of injury can occur, as we see below.

Case 3. Whilst we are awaiting Case 1 or Case 2 above, F can injure us by permitting $g \geq y(1)$ but with $\wedge(g, s(3)) < m(e, y(1), s(3))$ and $g > m(y(1))$. The problem is that perhaps we might force the restraint associated with N_e to infinity by constantly enumerating below $m(e, y(1), t)$. As N_e controls the enumeration of such numbers, our action is as follows. If F so permits g and $\wedge(g, t) < u(e, y(1), s(3))$ then enumerate $\wedge(g, t)$ into C. Do so for such g until a stage $\hat{t}$ occurs where $u(e, y(1), \hat{t}) = \hat{m}(e, y(1), \hat{t})$. Now if such $\wedge(g, t)$ occurs enumerate $\wedge(g, t)$ into C only if $\wedge(g, t) \leq \hat{m}(e, y(1), \hat{t})$. Otherwise, enumerate it into B. It can be seen that this device ensures that if Cases 1 and 2 above don't occur then N_e's restraint for the sake of $y(1)$ is finite.

The reader should note that Case 3 is compatible with Case 1 (and of course Case 2) since in Case 1, $\wedge(h, s(3))$ must be smaller than any $\wedge(g, t)$ of Case 3. That is, if while we are waiting for an e-expansionary stage (after we enumerate $\wedge(g, t)$), we see case 1 pertain then $h < g$ and we can apply Case 1 (and, for example, extract $\wedge(g, t)$ from C).

6 (*Injury after Case* 1). Suppose Case 1 occurs and so we create a disagreement at $z(1)$. As we have $C_t[s(1)] = C_{s(1)}[s(1)]$ we can protect this disagreement from all $h \geq y(1)$ (i.e., $\wedge(h, u)$) by enumerating such $\wedge(h, u)$ into C and not B—as $\wedge(h, u) \geq \wedge(h, s(1)) \geq s(1)$. That is—for $u > t$—if $h \in F_u - F_{u-1}$, $h \geq y(1)$ and $\wedge(h, u) < m(e, y(1), s(2)) \geq m(e, y(1), u)$, then enumerate $\wedge(h, u)$ into C_{u+1} and *not* B.

For such *injuries*, $\wedge(h, u) \geq \wedge(h, s(3))$, in fact, by kicking. The effect of this strategy is, like step 5, finite.

The only problem that occurs is if some $y(1, u) = \hat{y} = \hat{y}(1) < y(1)$ occurs in $F_{u+1} - F_u$ for some least $u \geq t$, after we have created a disagreement at $z(1)$. In this case our action is to enumerate $\wedge(\hat{y}, s(1)) = \wedge(\hat{y}, u)$ into both sides B and C and wait for the computations to recover. This is *injury after Case 1*, and we treat it like Case 2. That is, we return to step 3 in the cycle for $z(1)$ with $y(1, u)$ in place of $y(1, s(2))$.

7 (*The outcomes*). Because of the above our outcomes can only be outcome d (a win with finite effect by lack of recover), an outcome where we get to preserve a disagreement, or we get stuck awaiting a permission. Counting the number of failed permissions, we can see that in this last (inductive) outcome we must have some $y(1) = \lim_s y(1, s) \geq e$ such that for all x with $e \leq x \leq y(1)$ we have $D_{g(x)} \cap F \neq H \cap D_{g(x)}$. Call this outcome 1.

The cycle for subsequent $z(j)$.

Whilst it appears that outcome 1 (i.e., pending permission) seems correct for $\{z(j, s)\colon j \leq i\}$, we shall search for a $z(i + 1, s)$ at which we can make a disagreement.

For this we will begin a cycle for such $z(i+1, s)$ precisely as above. If s was the stage where it appeared that the cycle for $z(i, s)$ got stuck, the simplest way to mesh the strategies for $z(i+1, s)$ and $z(i, s)$ is to agree to cancel the cycle for $z(i+1, s)$ (and perhaps later pick a $z(i+1, t) \neq z(i+1, s)$) should F permit some h which the cycle for $z(i, s)$ forces us to enumerate $\wedge(h, s)$ in some special way (i.e., $\wedge(h, s) \leq m(e, z(i, s), s)$ and so $\wedge(h, s)$ falls within the "scope" of 1-6 of the cycle for $z(i, s)$). Again this injury affects the "$D_{g(x)} \cap F \neq D_{g(x)} \cap H$" strategy but as in the cycle for $z(1)$ and §2 these injuries can occur at most $< |D_{g(x)}|$ times for each x. To see this, it really suffices to argue for $z(1)$ and $z(2, s)$, since the general result uses the same technique, plus induction.

Thus suppose we are within a cycle for $z(1)$ for the sake of $y(1, s)$ as above. We can begin a cycle for $z(2, t)$ at $t \geq s$ predicted on the belief that this cycle is now permanently "hung", that is in particular for all h if $\wedge(h, s) < m(e, z(1), s)$ then $F[h] = F_s[h]$ (and moreover $F_s[m(y(1))] = F[m(y(1))]$. As a consequence we believe that for all x with $e \leq x \leq y(1)$,

$$D_{g(x)} \cap F_s = D_{g(x)} \cap F \neq D_{g(x)} \cap H_s = D_{g(x)} \cap H.$$

Whilst this belief is valid, we begin an inductive strategy on some $z(2, t)$. To get this we'll need stages $t(1)$ and $t(2)$ which were e-expansionary, $t(1) = ls(e, t(2)) > s$ where the conditions above pertain and we can use $t(1)$ and $t(2)$ to get a disagreement. Thus, for some $\hat{z} = z(2, s)$ we have

(i) $\Phi_{e, t(2)}(C_{t(2)}; \hat{z}) \neq \Phi_{e, t(1)}(C_{t(1)}; \hat{z})$, and
(ii) $\Phi_{e, t(2)}(B_{t(2)}; \hat{z}) \neq \Phi_{e, t(1)}(B_{t(1)}; \hat{z})$.

Yet the cycle for $z(1)$ has not acted. This means that some least $\hat{y}(1, s)$ has entered (or left)

$$(C_{t(2)} \cup D_{t(2)}) - (C_{t(1)} \cup D_{t(1)})$$

as in (3.2), (3.3) or (3.4). This $\hat{y}(1, t(1))$ must have

$$\wedge(h, t(1)) > m(e, z(1), t(1)) = m(e, z(1), s),$$

and we can simply make

$$D_{g(x)} \cap F_{t(2)} \neq D_{g(x)} \cap H_{t(2)+1}$$

for all x with $x \leq \hat{y}(1, t(1))$ by simply making them different on those x with $y(1, t(1)) = y(1, s) < x \leq \hat{y}(1, t(1))$.

Obviously there is no problem if our belief about $z(1)$ (i.e. being "hung") is correct. Note that we might need a new $z(2, t)$ and can injure $D_{g(x)}$ only if F permits some h with $h \leq x$ (i.e., assuming that we fail to meet N_e, the "F permitting $m(y(1, s))$ helpfully" will eventually be undone by F permitting some $y(1, t) < y(1, s)$). This can only occur $< x < |D_{g(x)}|$ times.

Thus, in general if N_e fails we will get $\lim_s z(i, s) = z(i)$ existing and

$$(\forall x \geq e)(D_{g(x)} \cap F \neq D_{g(x)} \cap H)$$

which would contradict the array nonrecursiveness of $\mathbf{f}$.

The general construction is to implement the above via a standard finite injury argement (as the π_2 outcome is inactive). Such details are routine and we leave them to the reader ■

The last result we will prove is:

(3.5) THEOREM. *There exists an r.e. degree* $\mathbf{a}$ *such that for all* $\mathbf{b}$ *with* $\mathbf{b} > \mathbf{a}$ *or* $\mathbf{b} \leq \mathbf{a}, \mathbf{b}$ *is the top of a diamond in the* Δ_2 *degrees.*

Proof. To prove (3.5) it will suffice to construct an r.e. array nonrecursive set A such that A is *strongly atomic* (as in [8], A strongly atomic means that whenever $A = A_1 \cup A_2$ is an r.e. splitting of A, i.e., $A_1 \cap A_2 = \emptyset, A_1 \cup A_2 = A$, then $\deg(A_1) \cap \deg(A_2) = \mathbf{0}$) and such that for all r.e. $B \leq_T A, B \leq_W A$. Theorem (3.5) will then follow since for all r.e. C, if $C_T \geq A$ then C is array nonrecursive (by [6]). Let $B \leq_T A$. Then $B \leq_W A$. Hence Sacks splitting A as $A_1 \cup A_2$ with $B \not\leq_T A_i$, we see $B = B_1 \cup B_2$ with $B_i \leq_W A_i$. (This follows since the W-degrees are a distributive upper semilattice—see e.g. [8], [14].) Then

$$\deg(B_1) \cap \deg(B_2) = \mathbf{0} \text{ and } \deg(B_1) \oplus \deg(B_2) = \deg(B).$$

To construct such an A we need to meet the requirements (for $\mathbf{Q} = \{D_{g(x)}: x \in \omega\}$ as before)

$$P_e: (\exists x)(D_{g(x)} \cap W_e = A \cap D_{g(x)}),$$
$$N_e: \text{If } V_e \cup U_e = A$$

and

$$\Phi_e(V_e) = \Phi_e(U_e) = f \text{ total} \Rightarrow f \text{ recursive},$$
$$R_e: \Gamma_e(A) = Q_e \Rightarrow Q_e \leq_W A.$$

Here (V_e, U_e, Φ_e) is a standard enumeration of all triples consisting of 2 *disjoint* r.e. sets and a reduction, and (Γ_e, Q_e) is an enumeration of pairs consisting of a reduction and an r.e. set. As the satisfication of the above requirements combines several known strategies, we really need to only sketch the details. To meet the P_e we attempt—for some follower "block" $D_{g(x)}$—to ensure that $W_e \cap D_{g(x)} = A \cap D_{g(x)}$ by enumeration of $D_{g(x)} \cap W_e$ into A. (Thus if $D_{g(x)}$ is a "stable" block then P_e can receive attention at most $|D_{g(x)}|$ times via x before it is met. Here by "stable" we mean a block that the strategy eventually settles on.)

To meet the N_e as in a minimal pair argument we need to ensure that at most one side of a $\Phi_{e,s}(V_{e,s}; x) = \Phi_{e,s}(U_{e,s}; x)$-computation changes between e-expansionary stages. Thus at an e-expansionary stage, we shall enumerate at most *one* element (i.e., below $u(\Phi_{e,s}(V_{e,s}; x))$ into A and then raise $r(e, s) = s$ until the next e-expansionary stage. As U_e and V_e are disjoint, this element can enter at most one side and the other side holds the computation. This strategy coheres with the P_j for $j \geq e$ in the usual way. If we are currently working on $D_{g(x)}$ for the sake of P_j, and at stage s, $r(e, s) > \hat{m}(x)$ we will begin a new strategy for P_j-guessing "$l(e, s) \nrightarrow \infty$"; i.e., the e-computations don't recover—on (e.g.) $D_{g(s)}$. Here $\hat{m}(p) = \min\{y\colon y \in D_{g(p)}\}$. If the e-computations recover, we abandon this strategy, but go back to $D_{g(\langle j, x\rangle)}$. This is all a standard π_2 procedure.

Finally to meet the R_e we use a confirmation procedure as in (e.g.) [5]. Let

$$L(e, s) = \max\{x\colon (\forall y < x)(\Phi_{e,s}(A_s; y) = Q_{e,s}(y))\}.$$

For a block D_z guessing that $L(e, s) \nrightarrow \infty$ at any stage we see $L(e, s) > \max\{L(e, t)\colon t < s)$, we cancel D_z. For a block $D_{g(x)}$ following P_j for $j \geq e$ guessing that $l(e, s) \to \infty$ at the first stage we see $L(e, s) > m(g(x))$ declare $D_{g(x)}$ (as a block) to be e-confirmed, and cancel all (assignments of) follower blocks $D_{g(y)}$ for $y > x$ (these have lower priority). We then only assign blocks D_z to requirements P_k after stage s if $z > s$ so that $\hat{m}(z) > s$. Also we assign in order of priority and ensure that if D_{z_1} is assigned to P_k and D_{z_2} to P_t and $k > t$ then $z_2 < z_1$ (i.e., assuming, of course, this version of P_t on the tree has a guess compatible with P_k). The effect of this is that after stage s for all $z < L(e, s)$, if $Q_{e,s}[z] \neq Q_e[z]$ then $A_s[m(x)] \neq A[m(x)]$. The other action is to ensure that if ever P_j requires attention via $D_{g(x)}$ then we shall *cancel* $D_{g(z)}$ for all z with $x < z \leq s$. This procedure occurs only $< |D_{g(x)}|$ times. Note that if D_z is assigned to P_k for some stage t *after* P_j receives attention, but *before* the next e-expansionary stage r (i.e., where $L(e, q) > \max\{L(e, \hat{q})\colon \hat{q} < q\}$) then at stage r, D_z will be cancelled as a follower of P_k).

From the above we see that if p is the least e-expansionary stage where $A_p[s] = A[s]$ then it follows that $A_p[p] = A[p]$ and hence $Q_{e,p}[z] = Q_e[z]$ for all $z \leq L(e, s)$. Hence $Q_e \leq_W A$.

We believe that most readers could now easily fill in the details of the above argument as they are, by now, fairly routine. Nevertheless, for completeness we shall now give some formal details.

Elements of $2^{<\omega}$ are called guesses with $\leq_L$ representing lexicographic ordering with $0 \leq_L 1$. We devote σ to N_e if $|\sigma| = 2e$ and σ is devoted to R_e if $|\sigma| = 2e + 1$. λ denotes the empty guess.

(3.6) DEFINITION. (a) A stage s is called a *σ-stage* by induction on $|\sigma|$ as follows

(i) Every stage s is a λ-stage

(ii) If s is a τ-stage and $|\tau| = 2e$ then we say s is a $\tau\hat{\ }0$-stage if

$$l(e, s) > \max\{l(e, t) \colon t \text{ is a } \tau\text{-stage and } t < s\}$$

where

$$l(e, s) = \max\{x \colon (\forall y < x)\{(V_{e,s} \cup U_{e,s})(y) = A_s(y) \ \& \ \Phi_{e,s}(V_{e,s}; y) = \Phi_{e,s}(U_{e,s}; y)]\}.$$

If s is not a $\tau\hat{\ }0$-stage then s is a $\tau\hat{\ }1$-stage.

(iii) If s is a τ-stage and $|\tau| = 2e + 1$ then if

$$L(e, s) > \max\{L(e, t) \colon t \text{ is } \tau\text{-stage and } t < s\}$$

we say s is a $\tau\hat{\ }0$-stage. Otherwise s is a $\tau\hat{\ }1$-stage.

(b) At stage s, the unique σ with $|\sigma| = s$ and s a σ-stage is denoted by σ_s.

(c) We say P_e requires attention at stage s if one of the following options holds

(3.7) P_e has a follower block $D_{g(x)}$ with guess $\subset \sigma_s$ and $A_s \cap D_{g(x)} \neq W_{e,s} \cap D_{g(x)}$.

(3.8) P_e has no follower block with guess $\subset \sigma_s$.

Construction, stage $s + 1$.

Step 1. Compute σ_s. Initialize all $\tau \not\leq_L \sigma_s$. That is, cancel all follower assignments guesses τ.

Step 2. *Confirmation.* Find the follower block $D_{g(x)}$ *with the highest priority guess τ such that, for some $\sigma < \tau$ with $\sigma < \sigma_s$ we have*

(i) $L(e, s) > \hat{m}(g(x))$ (where $|\sigma| = 2e + 1$);

(ii) $D_{g(x)}$ is not σ-confirmed.

If such x exists, declare $D_{g(x)}$ as γ-confirmed for all $\gamma < \sigma_s$ with $\gamma < \tau$ and $L(f, s) > \hat{m}(g(x))$ where $|\gamma| = 2f + 1$. Cancel all assignments of $D_{g(y)}$ with guess ρ for $\rho \not\leq_L \tau$. This step is vacuous if no such x exists.

Step 3. Find the least e such that P_e requires attention. Adopt the first case below to pertain, and say σ receives attention where $\sigma \subset \sigma_s$ and $|\sigma| = 2e + 1$.

Case 1. (3.7) holds. We claim

$$A_s \cap D_{g(x)} \subseteq W_{e,s} \cap D_{g(x)}. \tag{3.9}$$

Assuming (3.9) holds—as we later verify—find $z = (\mu y)[y \in (W_{e,s} \cap D_{g(x)}) - (A_s \cap D_{g(x)})]$. Enumerate z into $A_{s+1} - A_s$. Initialize all $\tau \not\leq_L \sigma$.

Case 2. (3.8) holds. Appoint $D_{g(s)}$ as a follower block of P_e with guess σ. Initialize all $\tau \not\leq_L \sigma$.

End of construction

Let β denote the leftmost path. That is $\beta \in [2^{<\omega}]$ is defined inductively via $\lambda \in \beta$. If $\sigma \subset \beta$ then $\sigma^\frown 0 \subset \beta$ iff $(\exists^\infty s)(s$ is a $\sigma^\frown 0$-stage) otherwise $\sigma^\frown 1 \subset \beta$.

(3.10) LEMMA. *Let $\sigma \leq_L \beta$ with $|\sigma|$ odd. Then:*

(i) *At all stages s, claim (3.9) is true of any $D_{g(x)}$ following P_e where $|\sigma| = 2e + 1$.*

(ii) *σ receives attention only finitely often.*

(iii) *If $\sigma \subset \beta$ then P_e is met at σ.*

(iv) *If $D_{g(x)}$ is any block with guess σ then $D_{g(x)}$ is τ-confirmed for $\tau \subset \sigma$ only finitely often.*

Proof. We prove the above by simultaneous induction. It is clear that (i) is true for all σ devoted to P_e since, when assigned, $D_{g(x)} \cap A_s = \emptyset$ and members of $D_{g(x)}$ are only added to A due to Step 3, Case 1. This only happens if (3.8) pertains, at which time we add members of $W_{e,s} \cap D_{g(x)}$ (one at a time), keeping $D_{g(x)} \cap A_s \subseteq W_{e,s} \cap D_{g(x)}$. Thus (i) holds.

We verify (ii), (iii) and (iv) by simultaneous induction. Let $\sigma \subset \beta$. We verify (ii), (iii) and (iv) for σ assuming them for all $\tau \leq_L \sigma$ with $\tau \neq \sigma$, (this is all that is necessary, since we are left of σ finitely often). Go to a stage s_0 where for all $\tau \leq_L \sigma$ with $\tau \neq \sigma$ and all $s > s_0$,

(i) τ does not receive attention at stage s,

(ii) $\sigma_s \not\leq_L \sigma$, and

(iii) no follower with guess τ is ever again γ-confirmed for any $\gamma \leq_L \sigma$.

As we initialize, we might as well suppose P_e has no follower block with guess σ. By construction at the next σ-stage $t \geq s_0$, P_e is given $D_{g(t)}$ as a follower with guess σ. This block is uncancellable. Because of this, σ will receive attention at most $|D_{g(t)}|$ times after stage t, whenever (3.8) holds. Therefore P_e is met and (iv) follows similarly. ∎

(3.11) LEMMA. *Let $\sigma\,^\wedge 0 \subset \beta$ with $|\sigma| = 2e + 1$. Then if $\Gamma_e(A) = Q_e$, $Q_e \leq_W A$.*

Proof. Let s_0 be a stage good for $\sigma\,^\wedge 0$ in the sense that all higher priority $\tau \leq_L \sigma\,^\wedge 0$ has ceased receiving attention etc. as in (3.10). We show how to compute $Q_e[z]$ for any z. Find the least $\sigma\,^\wedge 0$-stage $s > s_0$ such that $L(e, s) > m(g(x))$ and $L(e, s + 1) > m(g(x))$ for all $x \leq z$ with $D_{g(x)}$ still alive. After stage s, it is the case that either $D_{g(x)}$ has guess $\tau \leq_L \sigma$ (and so $D_{g(x)}$ does not enter) or $D_{g(x)}$ is $\sigma\,^\wedge 0$-confirmed. Let s_1 be the least $\sigma\,^\wedge 0$-stage with $s_1 > s$ such that $A_{s_1}[s] = A[s]$. We claim that $Q_{e,s}[x] = Q_e[x]$. The principal claim we need for this is to show that if t is a $\sigma\,^\wedge 0$-stage with $t \geq s$ then there are no follower blocks $D_{g(y)}$ alive with

$$(3.12) \qquad s \leq \hat{m}(g(y)) \leq u(\Gamma_{e,t}(A_t; x))$$

But this is easy to see as follows. The claim is clearly true at stage s. Now if no number $\leq s$ alive at stage s enters A, then $u(\Gamma_{e,s}(A_s; x)) = u(\Gamma_e(A; x))$. If any number from $D_{g(n)} \leq s$ enters, it cancels all $D_{g(y)}$ for $y > n$. Any number appointed between $s\,^\wedge 0$-stages has guess $\gamma \not\leq_L \sigma\,^\wedge 0$ and hence γ is cancelled at σ-stages. It follows that if n is least with numbers from $D_{g(n)}$ entering between stages s and t, then at stage t all numbers between $m(g(n))$ and t are cnacelled. Thus (3.12) follows. ∎

It is easy to see that if $\Gamma_e(A) = Q_e$ then $\sigma\,^\wedge 0 \subset \beta$ where $|\sigma| = 2e + 1$. It remains to verify the N_e. This is absolutely standard.

(3.13) LEMMA. *N_e is met.*

Proof. Suppose that $\Phi_e(U_e) = \Phi_e(V_e) = f$ total and $V_e \cup U_e = A$. Let $\sigma \subset \beta$ with $|\sigma| = 2e$. Then $\sigma\,^\wedge 0 \subset \beta$. Let s_0 be a stage good for σ. To compute $f(z)$ find the least $\sigma\,^\wedge 0$-stage $s = s(z) > s_0$ such that $l(e, s) > z$. We claim that

$$(3.14) \quad \Phi_{e,s}(U_{e,s}; z) = \Phi_{e,t}(U_{e,t}; z) \quad \text{or} \quad \Phi_{e,s}(V_{e,s}; z) = \Phi_{e,t}(V_{e,t}; z).$$

To see this, if (3.14) is to fail then two numbers below the use must enter the relevant sides between $\sigma\,^\wedge 0$-stages. Now at stage s there are no numbers alive $< s$ that can possibly enter A except those with guesses $\gamma \supset \sigma\,^\wedge 0$. Then by construction at a $\sigma\,^\wedge 0$-stage u only one number can enter A. All numbers to enter A between u and the next $\sigma\,^\wedge 0$-stage $u_1 > u$ must exceed u as they are appointed after stage u, and so must exceed both of the uses $u(\Phi_{e,u}(U_{e,u}; z))$ and $u(\Phi_{e,u}(V_{e,u}, z))$. Therefore, as U_e and V_e are disjoint, at most one side can change between $\sigma\,^\wedge 0$-stages. This gives (3.14) and the lemma. ∎

(3.15) COROLLARY. *The degree* **a** *of* (3.5) *can be low* (*and one exists below each promptly simple degree*).

Proof. The argument easily blends with lowness for the usual reasons; and similarly prompt permitting may be applied. ■

We mention (3.15) only because we don't know the possible jumps of such **a**. A slightly trickier argument based on the Downey-Jockusch construction (in [5]) of an r.e. 1-*topped* degree **a** that was strongly atomic will construct a 1-topped **a** as in (3.5). Here **a** is 1-topped if there is an r.e. set $A \in \mathbf{a}$ such that for all r.e. $B \leq_T A$, $B \leq_1 A$. The interest here is that it was shown in [5] that all 1-topped degrees are low$_2$-low.

This would then give:

(3.16) THEOREM. *The degree of* (3.5) *can be low$_2$-low.*

We would also like to point out that the techniques of (3.5)—or rather the observation that replacing the usual $\overline{A} \neq W_e$ with $(\exists x)(F_x \cap A = W_e \cap A)$ is compatible with most arguments where the lim inf on the true path is finite—leads us to the following observation (answering a question from [4]).

(3.17) THEOREM. (i) *There exists an array nonrecursive r.e. degree* **a** *such that for all r.e.* $\mathbf{b}_1, \mathbf{b}_2$ *with* $\mathbf{0} < \mathbf{b}_1, \mathbf{b}_2 < \mathbf{a}, \mathbf{b}_1 \cap \mathbf{b}_2 \neq \mathbf{0}$.

(ii) *Consequently, there exist r.e. degrees that are tops of diamonds in the* Δ_2 *degrees but not the d-r.e. degrees.*

Proof. (i) Array nonrecursiveness easily blends (as described above) with Lachlan's nonbounding construction [11].

(ii) This follows from (i) since all nonzero d-r.e. degrees bound nonzero r.e. ones.

REFERENCES

1. K. AMBOS-SPIES, *Anti-mitotic recursively enumerable sets*, Z. Math. Logik Grund. Math. **31** (1985), 461–477.
2. K. AMBOS-SPIES and P. FEJER, *Degree-theoretic splitting properties of recursively enumerable sets*, J. Symbolic Logic **53** (1988), 1110–1137.
3. S.B. COOPER, *Degrees of unsolvability complementary between recursively enumerable degrees part I*, Ann. Math. Logic **4** (1972), 31–73.
4. R.G. DOWNEY, *D-r.e. degrees and the non-diamond theorem*. Bull. London Math. Soc. **21** (1989), 43–50.
5. R.G. DOWNEY and C.G. JOCKUSCH, *T-degrees, jump classes and strong reducibilities*, Trans. Amer. Math. Soc. **301** (1987), 103–136.
6. R.G. DOWNEY, C.G. JOCKUSCH and M. STOB, *Array nonrecursive sets and multiple permitting arguments*, Recursion Theory Week, Proceedings Oberwolfach, 1989 (ed. K. Ambos-Spies, G. H. Müller, G. E. Sacks), Springer-Verlag, New York, pp. 141–174.

7. R.G. Downey and T.A. Slaman, *Completely mitotic r.e. degrees*, Ann. Pur Appl. Logic **41** (1989), 119–152.
8. R.G. Downey and L.V. Welch, *Splitting properties of r.e. sets and degrees*, J. Symbolic Logic **51** (1986), 88–109.
9. R.L. Epstein, *Initial segments of the degrees below* $\mathbf{0}'$, Mem. Amer. Math. Soc., no. 241, Amer. Math. Soc., Providence, Rhode Island, 1981.
10. P. Fejer, *Embedding lattices with top preserved below non-*GL_2 *degrees*, Z. Math. Logik Grund. Math. **35** (1989), pp. 3–14.
11. A.H. Lachlan, *Bounding minimal pairs*, J. Symbolic Logic **44** (1979), 626–642.
12. R. E. Ladner, *Mitotic recursively enumerable sets*, J. Symbolic Logic **38** (1973), 199–211.
13. ______, *A completely mitotic non-recursive recursively enumerable degree*, Trans. Amer. Math. Socl. **184** (1973), 479–507.
14. R.E. Ladner and L. Sasso, *The weak truth table degrees of recursively enumerable sets*, Ann. Math. Logic **8** (1975), 429–448.
15. D. Posner, *The upper semilattice of degrees* $\leq \mathbf{0}'$ *is complemented*, J. Symbolic Logic **44** (1979), 626–642.
16. T.A. Slaman, *A recursively enumerable degree that is not the top of a diamond in the Turing degrees*, to appear.
17. R.I. Soare, *Recursively enumerable sets and degrees*, Springer Verlag, New York, 1987.

Victoria University of Wellington
Wellington, New Zealand