JUMPS OF MINIMAL DEGREES BELOW $\mathbf{0}'$

Rodney G. Downey, Steffen Lempp, and Richard A. Shore

Abstract. We show that there is a degree $\mathbf{a}$ REA in and low over $\mathbf{0}'$ such that no minimal degree below $\mathbf{0}'$ jumps to a degree above $\mathbf{a}$. We also show that every nonlow r.e. degree bounds a nonlow minimal degree.

Introduction.

An important and long-standing area of investigation in recursion theory has been the relationship between quantifier complexity of the definitions of sets in arithmetic as expressed by the jump operator and the basic notion of relative computability as expressed by the ordering of the (Turing) degrees. In this paper we are concerned with an aspect of the general problem of characterizing the range of the jump operator on various classes of degrees. The first such result was the completeness or jump inversion theorem of Friedberg [1957].

Theorem (Friedberg Jump Inversion). *If $\mathbf{c} \geq \mathbf{0}'$ then there is an $\mathbf{a}$ such that $\mathbf{a}' = \mathbf{c}$.*

As $\mathbf{a}'$ is obviously at least $\mathbf{0}'$ for every degree $\mathbf{a}$, this result says that every "possible" degree $\mathbf{c}$ (i.e., every degree not ruled out on trivial grounds) is the jump of some degree $\mathbf{a}$. A number of other important jump inversion theorems of this sort followed Friedberg's. In particular Shoenfield [1959] and Sacks [1963] characterized the jumps of the degrees below $\mathbf{0}'$ and of the r.e. degrees, respectively, as all "possible" degrees:

Theorem (Shoenfield Jump Inversion). *If $\mathbf{c} \geq \mathbf{0}'$ and $\mathbf{c}$ is r.e. in $\mathbf{0}'$ then there is a degree $\mathbf{d} \leq \mathbf{0}'$ such that $\mathbf{d}' = \mathbf{c}$.*

Theorem (Sacks Jump Inversion). *If $\mathbf{c} \geq \mathbf{0}'$ and $\mathbf{c}$ is r.e. in $\mathbf{0}'$ then there is an r.e. degree $\mathbf{d}$ such that $\mathbf{d}' = \mathbf{c}$.*

Our concern in this paper is the problem of jump inversion into the minimal degrees below $\mathbf{0}'$. The general problem for the degrees as a whole was solved by Cooper [1973]:

1991 *Mathematics Subject Classification.* 03D25.

Key words and phrases. minimal degree, Turing jump, low degree.

Research partially supported by NSF Grants DMS-9100114 (Lempp), DMS-9344740 (Shore), a U.S.-New Zealand Binational Cooperative Research grant including NSF INT 90-20558 (Downey, Lempp and Shore), ARO through ACSyAm at MSI, Cornell University, DAAL-03-91-C-0027 (Downey and Shore), and IGC of Victoria University, Wellington (Downey).

Theorem (Cooper Jump Inversion). *If* $\mathbf{c} \geq \mathbf{0}'$ *then there is a minimal degree* $\mathbf{m}$ *such that* $\mathbf{m}' = \mathbf{c}$.

The natural conjecture at this point would have been that the Shoenfield jump inversion theorem could be extended analogously, i.e., if $\mathbf{c} \geq \mathbf{0}'$ is r.e. in $\mathbf{0}'$ then there should be a minimal $\mathbf{m} \leq \mathbf{0}'$ with $\mathbf{m}' = \mathbf{c}$. Cooper [1973], however, refuted this by showing that no *high* degree $\mathbf{m}$ below $\mathbf{0}'$ (i.e., no degree with $\mathbf{m}' = \mathbf{0}''$) can be minimal. The only known positive result at the time was that there were minimal $\mathbf{m} < \mathbf{0}'$ with $\mathbf{m}' = \mathbf{0}'$. (This follows from the existence of a minimal degree below any r.e. degree $\mathbf{a}$ (Yates [1970]) by taking $\mathbf{a}$ to be low (i.e., $\mathbf{a}' = \mathbf{0}'$).) Indeed, the minimal degrees $\mathbf{d}$ constructed in the proof of the Cooper jump inversion theorem all had the property that $\mathbf{m}' = \mathbf{m} \vee \mathbf{0}'$ ($\mathbf{m}$ is in $\mathbf{GL}_1$ or *generalized low*$_1$). Answering a question from Yates [1974], Sasso [1974] proved that this is not always the case by constructing a minimal $\mathbf{m} < \mathbf{0}'$ with $\mathbf{m}' > \mathbf{0}'$.

In the early 70's Jockusch conjectured (see Sasso [1974, p.573], Yates [1974, p. 235]) that the jumps of minimal degrees $\mathbf{m} < \mathbf{0}'$ are precisely the set of $\mathbf{c} \geq \mathbf{0}'$ with $\mathbf{c}$ r.e. in $\mathbf{0}'$ and low over $\mathbf{0}'$, i.e., $\mathbf{c}' = \mathbf{0}''$. Jockusch and Posner [1978] eventually proved the (remarkable) half of this conjecture.

Theorem (Jockusch and Posner [1978]). *Every minimal degree* $\mathbf{m}$ *is in* $\mathbf{GL}_2$, *i.e.,* $(\mathbf{m} \vee \mathbf{0}')' = \mathbf{m}''$. *In particular, every minimal* $\mathbf{m} \leq \mathbf{0}'$ *is low*$_2$, *i.e.,* $\mathbf{m}'' = \mathbf{0}''$.

The other half of Jockusch's conjecture, that the jumps of minimal degrees $\mathbf{m} \leq \mathbf{0}'$ include all those $\mathbf{c} \geq \mathbf{0}'$ r.e. in $\mathbf{0}'$ with $\mathbf{c}' = \mathbf{0}''$, has remained an open problem.

In this paper we refute this attractive conjecture by exhibiting an entire upward cone of such degrees which contains no jump of a minimal degree below $\mathbf{0}'$:

Theorem 1.1. *There is a degree* $\mathbf{a}$ *recursively enumerable in and above* $\mathbf{0}'$ *with* $\mathbf{a}' = \mathbf{0}''$ *such that* $\mathbf{a}$ *is not recursive in the jump of any minimal degree below* $\mathbf{0}'$.

A similar but (in terms of determining the range of the jump on the minimal degrees below $\mathbf{0}'$) somewhat weaker result has been independently (and slightly earlier) obtained by Cooper and Seetapun (personal communcation[1]):

Theorem (Cooper and Seetapun). *There is a degree* $\mathbf{a}$ *recursively enumerable in and above* $\mathbf{0}'$ *with* $\mathbf{a}' = \mathbf{0}''$ *such that if* $\mathbf{d} \leq \mathbf{0}'$ *and* $\mathbf{a}$ *is r.e. in* $\mathbf{d}$ *then there is a* 1*-generic degree below* $\mathbf{d}$.

We have not seen Cooper and Seetapun's proof but they have told us that it is a $\mathbf{0}'''$-argument like the proof of Lachlan's nonbounding theorem as presented in Soare [1987, Chapter XIV]. Our construction is of a considerably simpler type. Indeed, it is really a finite injury argument relative to K (as one might expect in building a low r.e. degree $\mathbf{a}$ relative to K). As we cannot actually deal with potential minimal degrees $\mathbf{d} < \mathbf{0}'$ in a construction recursive in K, we use recursive approximation procedures for K and A which turn the construction into a $\mathbf{0}''$-tree argument.

We also prove that the theorem fails if we try to replace upward cones by downward ones.

[1]Recently, Cooper has claimed a full characterization of this class of degrees as the class of what he calls the almost Δ_2-degrees.

Theorem 2.1. *If D is any r.e. set with $\emptyset' <_T D'$, then there is a set M of minimal degree with $M \leq_T D$ and $\emptyset' <_T M' \leq_T D'$.*

Corollary 2.3. *If $\mathbf{c} > \mathbf{0}'$ is r.e. in $\mathbf{0}'$, then there is a minimal degree $\mathbf{m} < \mathbf{0}'$ such that $\mathbf{0}' < \mathbf{m}' \leq \mathbf{c}$.*

Thus the range of the jump operator on the minimal degrees below $\mathbf{0}'$ cannot be characterized simply in terms of the jump classes of degrees r.e. in and above $\mathbf{0}'$.

We finally include sketches of the proofs of two old unpublished related results by the third author concerning cone avoiding by the jumps of minimal degrees below $\mathbf{0}'$:

Theorem 3.1. *If $\mathbf{c}$ is r.e. in and above $\mathbf{0}'$ and $\mathbf{c}' = \mathbf{0}''$ then (uniformly in the information) there is a minimal degree $\mathbf{a} < \mathbf{0}'$ with $\mathbf{a}' \not\leq \mathbf{c}$.*

Theorem 3.2. *There are minimal degrees $\mathbf{a}_0, \mathbf{a}_1 < \mathbf{0}'$ such that $\mathbf{a}_0' \cup \mathbf{a}_1' = \mathbf{0}''$.*

1. An upward cone with no jumps of minimal degrees.

In this section we will prove that there is a degree $\mathbf{a}$ r.e. in $\mathbf{0}'$ and low over K such that no minimal degree $\mathbf{m} < \mathbf{0}'$ jumps above it.

Theorem 1.1. *There is a degree $\mathbf{a}$ recursively enumerable in and above $\mathbf{0}'$ with $\mathbf{a}' = \mathbf{0}''$ such that $\mathbf{a}$ is not recursive in the jump of any minimal degree below $\mathbf{0}'$.*

We let $K = \{\langle x, y\rangle \mid \varphi_x(y)\downarrow\}$ be the complete r.e. set. Our plan is to define a Σ_2-set A via a recursive approximation so that the degree of $A \oplus K$ is the $\mathbf{a}$ of our theorem. There are two types of requirements that we must satisfy. The first is the standard lowness requirement relativized to K:

$$N : (A \oplus K)' \equiv_T K'.$$

As usual, N is divided up into infinitely many requirements

$$N_e : \text{preserve } \Phi_e(A \oplus K; e).$$

As our construction will be recursive, we can only expect to succeed modulo the K-correctness of our computations (and, of course, the outcomes of requirements of higher priority).

The second type of requirements, P, deal with the jumps of minimal degrees below $\mathbf{0}'$. We can list all candidates, Δ, for minimal degrees below $\mathbf{0}'$ by taking a list of all partial recursive $\{0,1\}$-valued functions $\Delta(x, s)$. If we let $\Delta(x) = \lim_{s\to\infty} \Delta(x, s)$, we have, by the limit lemma, a list which includes all sets recursive in K. One more application of the limit lemma tells us how to represent sets recursive in Δ', the jump of Δ. They are all those of the form $\Gamma = \lambda y \lim_{n\to\infty} \Gamma(\Delta, y, n)$ for some partial recursive functional Γ of the appropriate form. Thus we can express the remaining conditions of the theorem by the requirements $P_{\Delta,\Gamma}$ as $\Delta(x, s)$ and $\Gamma(\Delta, y, x)$ range over the appropriate classes of partial recursive functionals:

$P_{\Delta,\Gamma}$: If Δ and Γ are total functions and $\Gamma = A$ then there is a Z such that $0 <_T Z <_T \Delta$.

We divide each requirement $P_{\Delta,\Gamma}$ into two infinite sequences of subrequirements

$$R_{\Delta,\Gamma,\Psi} : Z \neq \Psi$$
$$S_{\Delta,\Gamma,\Psi} : \Psi(Z) \neq \Delta$$

where Ψ ranges over all partial recursive functionals (and Ψ is short for $\Psi(\emptyset)$).

If the *hypotheses of* $P_{\Delta,\Gamma}$ *hold*, Δ and Γ are total and $\Gamma = A$, then our plan is to define a recursive functional Λ so that setting $\Lambda(\Delta) = Z$ satisfies $R_{\Delta,\Gamma,\Psi}$ and $S_{\Delta,\Lambda,\Psi}$ for each functional Ψ. For each subrequirement we will have a number y called the *killing point* which we will alternately put into and take out of A to try to force a win (We will explain the precise mechanics of this later. For now just note that the killing point for the version of a requirement at node α will be (the code of) α.) Roughly speaking, the plan for $R = R_{\Delta,\Gamma,\Psi}$ is as follows: We put y into A and wait for an n such that $\Gamma(\Delta, y, n) = 1$. (If there is no such n, $\Gamma \neq A$.) When we get such an n, we *use* it by choosing a *follower* x and setting $Z(x) = 0 = \Lambda(\Delta, x)$ by an axiom from Δ of length $\gamma(y, n)$. We now wait for x to be *realized*, i.e., for $\Psi(x) = 0$. At this point, we remove y from A. If Δ changes on $\gamma(y, n)$ we can redefine $\Lambda(\Delta, x) = Z(x)$ to be 1 and win R. If not we can put y back into A and try again for a new n and x. The idea is that if we never get a win ($Z(x) \neq \Psi(x)$) we cycle through this scenario infinitely often each time getting an n with $\Gamma(\Delta, y, n) = 1$ via a Δ-correct computation. As we remove y from A infinitely often, $y \notin A^K$. As $y \notin A^K$ requires that $\lim_{n\to\infty} \Gamma(\Delta, y, n) = 0$, we have the desired failure of the hypotheses of $P_{\Delta,\Gamma}$.

Note that if $x \in \omega^{[i]}$ and the i^{th} requirement is not an $R_{\Delta,\Gamma,\Psi}$ for any Ψ then $Z(x) = 0$ by a trivial axiom $\langle \emptyset, x, 0 \rangle$ which we put into Λ at stage 0.

The basic plan for satisfying $S = S_{\Delta,\Gamma,\Psi}$ is similar. We begin by putting y into A and waiting for an n such that $\Gamma(\Delta, y, n) = 1$ that we can *use*. We also want a Ψ-computation of the Δ use from Z, $\Psi(Z){\upharpoonright}\gamma(y, n) = \Delta{\upharpoonright}\gamma(y, n)$. At this point, we remove y from A. If we get a Δ change on $\gamma(y, n)$ we would hope to be able to restore $Z{\upharpoonright}\psi\gamma(y, n)$ to its previous value for a diagonalization $\Psi(Z){\upharpoonright}\gamma(y, n) \neq \Delta{\upharpoonright}\gamma(y, n)$. If we cannot do this, we return y to A and try again for a new n with $\Gamma(\Delta, y, n) = 1$ and $\Psi(Z) = \Delta \upharpoonright \gamma(y, n)$. We will have to argue that we can be prevented from eventually restoring Z to one of these older desired values only if there are infinitely many n with Δ-correct computations $\Gamma(\Delta, y, n) = 1$. As $y \notin A$ we will again have contradicted the hypotheses of $P_{\Delta,\Gamma}$.

The general structure of our argument is actually that of a finite injury argument (modulo the approximations to K). Although this may seem surprisingly simple, we are after all constructing (relative to K) a low r.e. set. General considerations as in Soare [5] then suggest that our argument should be a finite injury one (over K). As the actual construction must be recursive, our approximation procedure produces a standard tree structure for it.

We begin by listing all the requirements N_e, $R_{\Delta,\Gamma,\Psi}$ and $S_{\Delta,\Gamma,\Psi}$ in an ω-list Q_i. As usual, the tree structure will consist of sequences α of outcomes of the requirements Q_i, $i < |\alpha|$. Such a node α will be devoted to requirement $Q_{|\alpha|}$. We say that α is *associated with* $P_{\Delta,\Gamma}$ if $Q_{|\alpha|}$ is $R_{\Delta,\Gamma,\Psi}$ or $S_{\Delta,\Gamma,\Psi}$ for some Ψ. The possible outcomes of a requirement $R_{\Delta,\Gamma,\Psi}$ or $S_{\Delta,\Gamma,\Psi}$ are 0 and 1 depending on whether it believes $y \notin A$ or $y \in A$, respectively. A requirement N_e can have any element of ω as its outcome. It will be the use of $\Phi_e(A^K \oplus K; e)$ if it is believed to converge and 0 otherwise. To specify the actions and outcomes of the requirements more precisely we need to fix various approximation procedures.

Approximations: We have already fixed K as the complete r. e. set. We also fix some recursive one-one enumeration $k(s)$ of K. As usual, $K_s = \{k(t) \mid t < s\}$. We approximate Δ in the obvious way: $\Delta_s(x) = \Delta(x, s)$. We follow the convention of adding an s in brackets at the end of an expression to denote that we are using the s^{th} stage approximation to each set or functional in the expression. Thus $K(x)[]$ and $\Delta(x)[s]$ mean $K_s(x)$ and $\Delta_s(x)$. More interestingly, $\Gamma(\Delta, y, n) = 1[s]$ means that $\Gamma_s(\Delta_s, y, n)\downarrow = 1$. (To compute the functional Γ for s steps.) If we wish to denote the state of affairs at substage i of stage s we append "$[s, i]$" to the relevant expression. We also adopt the standard conventions that small Greek letters such as $\gamma(y, n)$ indicate the use of the functional denoted by the corresponding capital Greek letters $\Gamma(\Delta, y, n)$ and that uses are nondecreasing in both the input and stage of the approximation.

For each Δ and Γ we will enumerate axioms into a functional $\Lambda_{\Delta,\Gamma} = \Lambda$ to define $Z = \Lambda(\Delta)$. We can take the axioms to be of the form $\langle \sigma, x, j \rangle$ where σ is a binary string and, for $j = 0, 1$, $\Lambda(\Delta; x) = j \Leftrightarrow \exists \sigma(\langle \sigma, x, j \rangle \in \Lambda \;\&\; \sigma \subseteq \Delta)$. This defines the obvious approximations to $Z : Z_s(x) = j \Leftrightarrow \exists \sigma(\langle \sigma, x, j \rangle \in \Lambda_s \;\&\; \sigma \subseteq \Delta_s)$. Of course, we must make sure that axioms are enumerated so that Λ is consistent and that $\Lambda(\Delta)$ is total if Δ and Γ satisfy the hypotheses of $P_{\Delta,\Gamma}$.

The last approximation is the one to the Σ_2 set A that we are constructing. We begin with $A = \emptyset$ at stage 0. At various times during the construction we may put a number y into A or take it out. Our approximations to, or beliefs about, A at any point in the construction are the obvious ones: We believe that $A(x) = 1$ if the last action we took was to put x into A and $A(x) = 0$ otherwise. Our final Σ_2 set A is defined by $A(x) = 1 \Leftrightarrow$ there is a point in the construction at which we put x into A and then never take it out. We can now describe our construction.

The construction: We begin each stage s of the construction by declaring the empty sequence $\emptyset$ to be *accessible*. Stage s will now have s substages each of which starts with some node α being accessible. If α is the node accessible at the beginning of substage i, we remove β from A for every β to the right of α. If permitted, we may now act for requirement Q_i. In any case, we declare $\alpha\hat{\ }q$ to be accessible for some $q \in \omega$. If $i < s$, we proceed to substage $i + 1$. If $i + 1 = s$, we finish stage s and move on to stage $s + 1$. Our actions at substage i depend on the form of the requirement Q_i.

$Q_i = N_e$: Let r be the last α-*stage*, i.e., one at which α was accessible (0 if this is the first α-stage). Let $k(\alpha, s)$ be the least number enumerated in K since stage r, i.e., $k(\alpha, s) = \min(K_s - K_r)$. We say that there is an α-*believable* computation of $\Phi_e(A \oplus K; e)\downarrow$ if $\Phi_e(A \oplus K \restriction k(\alpha, s); e)\downarrow [s, i]$. If there is such a computation with use u we declare $\alpha\hat{\ }u$ to be *accessible*. If not, we declare $\alpha\hat{\ }0$ to be *accessible*.

If Q_i is of the form $R_{\Delta,\Gamma,\Psi}$ or $S_{\Delta,\Gamma,\Psi}$, the *killing point* for α is α itself, i.e., the natural number coding the sequence α. The dependence of α's actions on higher priority requirements in terms of its enumerating axioms in Λ is expressed in terms of its *dependency set*, $D(\alpha, s) = \{\langle \beta, m \rangle \mid \beta \leq \alpha \wedge \beta$ is associated with $P_{\Delta,\Gamma}$ and has used m at some point prior to substage $|\alpha|$ of stage $s\}$.

We begin substage i by having α *impose a global wait* for $\Gamma(\Delta, \beta, m)$ to be convergent for every $\langle \beta, m \rangle \in D(\alpha, s)$, i.e., no node $\gamma \geq \alpha$ associated with $P_{\Delta,\Gamma}$ can act at any stage $t \geq s$ unless $\Gamma(\Delta, \beta, m)\!\downarrow [t]$ for all $\langle \beta, m \rangle \in D(\alpha, s)$. If this condition is not met, we proceed directly to the terminal step (5) below of substage i. Otherwise, our actions depend on whether Q_i is $R_{\Delta,\Gamma,\Psi}$ or $S_{\Delta,\Gamma,\Psi}$.

$Q_i = R_{\Delta,\Gamma,\Psi}$: Step 1. If there is a marker x such that α *is satisfied by* x, i.e., $Z(x)\!\downarrow = 1$ and $\Psi(x)\!\downarrow = 0$ then we proceed to step 4 below. If not, we see if we *can satisfy* α via some marked x, i.e., we see if there is a marked x not controlled by any $\beta < \alpha$ such that $Z(x)\!\uparrow$. In this case we set $Z(x) = 1$ by enumerating an axiom $\langle \Delta_s \restriction m, x, 1 \rangle$ into Λ where m is the maximum of the length of any previous axiom for x in Λ and $u(\alpha, s) = \max\{\gamma(\beta, m)[s] \mid \langle \beta, m \rangle \in D(\alpha, s)\}$. (Note that all such $\gamma(\beta, m)$ are defined by *our global wait requirements.*) If so, go to step 4. Otherwise, let d be the number of times α has been taken out of A and proceed to step 2.

Step 2. If α is not in A, put α into A and *impose a global wait for an $n > d$, with* $\Gamma(\Delta, \alpha, n)\!\downarrow = 1$, i.e., no action can be taken for any $\beta \geq \alpha$ associated with $P_{\Delta,\Gamma}$ at any stage $t \geq s$ unless there is an $n > d$ with $\Gamma(\Delta, \alpha, n)\!\downarrow = 1[t]$ until α is removed from A. If at any $t \geq s$ we act for a requirement β on the basis of such a computation $\Gamma(\Delta, \alpha, n) = 1$, we say that α *uses n at t*. If this global wait condition is satisfied, we proceed to step 3. Otherwise, we go to step 5.

Step 3. If there is an unrealized follower x of α we proceed to step 4. If there is a realized follower x of α which is associated with a number $n > d$ and $\Gamma(\Delta, \alpha, n) = 1[s]$ then we *mark* x, remove α from A and proceed to step 4. If there is no such x but there is an $m > d$ not used by α such that $\Gamma(\Delta, \alpha, m) = 1[s]$ let n be the least such. We now *use n* by *associating it with a follower $x \in \omega^{[i]}$ of α* which is larger than any number mentioned so far. We set $Z(x) = 0$ by an axiom of length

$$u(\alpha, s) = \max\big(\{\gamma(\beta, m)[s] \mid \langle \beta, m \rangle \in D(\alpha, s)\} \cup \{\gamma(\alpha, n)[s]\}\big).$$

We say that x is *realized* at $t \geq s$ if $\Psi_t(x) = 0$, otherwise it is *unrealized.* In any case, we now go to step 4.

Step 4. We have α *take control* of every $x < s$, $x \in \omega^{[i]}$ which is not currently controlled by any $\beta < \alpha$. (Of course, if $\delta > \alpha$ had control of x before, it no longer does.) If $x \in \omega^{[i]}$ is now controlled by some $\delta \leq \alpha$ (which is necessarily associated with $P_{\Delta,\Gamma}$), δ first took control of x at $t \leq s$ and $Z(x)$ is now undefined (i.e., $\sigma \not\subseteq \Delta_s$ for every $\langle \sigma, x, j \rangle$ now in Λ), we redefine $Z(x)$ to be its value at the last δ-stage (0 if there is none) via an axiom of length the maximum of that of any axiom previously defining $Z(x)$ and $u(\delta, t, s)$ where $u(\delta, t, s) = \max\{\gamma(\beta, m)[s] \mid \langle \beta, m \rangle \in D'(\delta, t)\}$ and $D'(\delta, t) = \{\langle \beta, m \rangle \mid \beta \leq \delta \wedge \beta$ is associated with the same $P_{\Delta,\Gamma}$ as $\delta \wedge \beta$ has used n at some point prior to the end of substage $|\delta|$ of stage $t\}$. We now go to step 5.

Step 5. If α is now in A, we let $\alpha^\frown 1$ be accessible. Otherwise, we let $\alpha^\frown 0$ be accessible. This ends substage i of stage s.

$Q_i : S_{\Delta,\Gamma,\Psi}$: Step 1. If α *is satisfied*, i.e., there is an x controlled by α such that $\Psi(Z;x)\downarrow \neq \Delta(x)[s]$, we go to step 4. Otherwise, we see if we *can satisfy* α, i.e., there is an n marked for α at some $r < s$ such that, for every $x < \psi(\gamma(\alpha,n))[r,i]$, $Z(x)[s,i]$ is either undefined or equal to $Z(x)[r,i]$. If so, α *takes control* of every $x < \psi(\gamma(\alpha,n))[r,i]$ with $x \in \omega^{[i]}$ such that Q_i is associated with $P_{\Delta,\Gamma}$ (no others can ever even appear to be in Z.) not controlled by some $\beta < \alpha$. (Of course, no $\beta > \alpha$ controls any such x from now on.) For each such x, we redefine $Z(x)$ to be $Z(x)[r,i]$ by an axiom of length $u(\alpha,s)$. (The relevant $\gamma(\beta,n)$ are all defined by our global wait requirements. Moreover, our procedures for redefining $Z(x)$ (in Steps 4) guarantee that $Z(x)\downarrow [s,i]$ for all x controlled by some $\beta < \alpha$.) We now go to step 4. If we cannot satisfy α, we let d be the number of times α has been removed from A and proceed to step 2.

Step 2. If α is not in A, put α into A and impose a global wait for an $n > d$ with $\Gamma(\Delta,\alpha,n) = 1$. If this condition is now satisfied we go to step 3. Otherwise we go to step 5.

Step 3. If there is an n marked for α such that $\Psi(Z;x)$ is undefined for some $x < \gamma(\alpha,n)[s]$, go to step 4. If not and there is an unmarked $n > d$ such that $\Gamma_1(\Delta,\alpha,n) = 1[s]$ and $\Psi(z)\restriction\gamma(y,n) = \Delta\restriction\gamma(\alpha,n)[s,i]$ then α *uses* and *marks* the least such n. We remove α from A and proceed to step 4.

Step 4. If x is controlled by some $\beta \leq \alpha$ and $Z(x)\uparrow\ [s,i]$, we redefine $Z(x)$ as we did for $R_{\Delta,\Gamma,\Psi}$.

Step 5. If α is now in A, $\alpha\widehat{\ }1$ is accessible. Otherwise, $\alpha\widehat{\ }0$ is accessible. This ends substage i of stage s.

End of Construction.

We must now verify that the construction defines a set A which satisfies all the requirements N and $P_{\Delta,\Gamma}$. We begin by showing that there is a leftmost path f on the tree T of nodes α which are ever accessible and that $(A \oplus K)'$ is determined by the outcome along f. Remember that an infinite path f on T is the *leftmost path* on T if $\forall \alpha \subset f$ (α is the leftmost node on level $|\alpha|$ of T which is accessible infinitely often).

Lemma 1.2. *There is a leftmost path f on T and the outcomes along f are the true ones:*

i) If Q_i is $R_{\Delta,\Gamma,\Psi}$ or $S_{\Delta,\Gamma,\Psi}$ and $\alpha = f\restriction i$, then $\alpha \in A \Leftrightarrow f(\alpha) = 1$.

ii) If $Q_i = N_e$, then $\Phi_e(A \oplus K; e)\downarrow \Leftrightarrow f(i) = 1$.

Proof. We proceed by induction on the length of $\alpha \subset f$. Clearly $\alpha = \emptyset$ is the leftmost sequence of length 0 accessible infinitely often. Suppose by induction that $|\alpha| = i$ and α is the leftmost node of length i accessible infinitely often. Let s_0 be such that no node $\beta <_L \alpha$ is accessible at any $s \geq s_0$.

i) If Q_i is $R_{\Delta,\Gamma,\Psi}$ or $S_{\Delta,\Gamma,\Psi}$ and α is accessible at s, then $\alpha\widehat{\ }1$ is accessible if α is in A and $\alpha\widehat{\ }0$ is accessible otherwise. As $\alpha \in A$ iff it is in A from some point on, $\alpha\widehat{\ }A(\alpha)$ is clearly the leftmost immediate successor of α which is accessible infinitely often.

ii) $Q_i = N_e$: We say an α-stage s is *K-true* if $K_s \restriction k(\alpha, s) = K \restriction k(\alpha, s)$. There are clearly infinitely many such stages. The outcomes of α at such stages are what really matter.

Sublemma 1.2.1. *If there is a K-true α-stage $s_1 \geq s_0$ at which there is an α-believable computation of $\Phi_e(A \oplus K; e)$ with use u, then this computation is $A \oplus K$-correct, i.e., $(A \oplus K) \restriction u = (A \otimes K) \restriction u[s, i]$ and $\alpha\widehat{\ }u$ is accessible at every α-stage $s \geq s_1$.*

Proof. First note that α's beliefs about $A(\beta)$ for $\beta < u$ and $\beta \not\supset \alpha$ are correct and return to their correct state at every α-stage $s \geq s_1$. For $\beta <_L \alpha$, this holds by our choice of s_0. For $\beta >_L \alpha$, it holds by our removing β from A at the beginning of substage i of every α-stage. For $\beta \subset \alpha$, it holds by induction. On the other hand, as long as $\alpha\widehat{\ }u$ is accessible, we cannot change $A(\beta)$ for any $\beta <_L \alpha\widehat{\ }u$. As $K_s \restriction u$ never changes by the definitions of a K-true α-stage and α-believability, we see that the current computation at s is α-believable at every α-stage. Of course, any $\beta \supseteq \alpha\widehat{\ }v$ for $v \geq u$ is itself bigger than u and so irrelevant. Thus $A(\beta)$ is never changed at $s \geq s_1$ for $\beta < u$ and $\beta \supset \alpha$ and so the computation is $A \oplus K$ correct as well. $\square$

Note that we have now proved that α has a leftmost successor which is accessible infinitely often: If the hypotheses of Sublemma 1.2.1 hold then $\alpha\widehat{\ }u$ is accessible at all sufficiently large α-stages. If not, $\alpha\widehat{\ }0$ is accessible at every K-true α-stage. Thus there is a leftmost path f on Γ whose outcomes for nodes $R_{\Delta,\Gamma,\Psi}$ and $S_{\Delta,\Gamma,\Psi}$ are correct, i.e., (i) holds. We conclude the verification of Lemma 1.2 by establishing the following:

Sublemma 1.2.2. *If $\Phi_e(A \oplus K; e)\downarrow$ with use u, then $\alpha\widehat{\ }u$ is accessible at all sufficiently large α-stages.*

Proof. The proof of Sublemma 1.2.1 shows that it suffices to prove that the correct computation is α-believable at some α-stage $s \geq s_0$ by which $K_s \restriction uK \restriction u$ and $k(\alpha, s) > u$. It also shows that α's beliefs are correct at all such stages for all $\beta \not\supset \alpha$. Let $\delta \subset f$ be of length u and suppose that δ is accessible at $s_1 \geq s_0$. By our results so far, ((i) holds), we know that $A \restriction u[s_1, s_1] = A \restriction u$. If s is the first α-stage after s_1, then no changes have been made on $A(\beta)$ for $\beta < \alpha$ and so $A \restriction u[s, i] = A \restriction u$ and the true computation of $\Phi_e(A \oplus K; e)\downarrow$ is α-believable as required. $\square$

Corollary 1.3. $(A \oplus K)' \equiv K'$.

Proof. Clearly $K' \leq (A \oplus K)'$. It thus suffices to show that we can compute $(A \oplus K)'$ from K'. Now, by Lemma 1.2, there is a leftmost path f on T. By definition, $f \leq K'$. As the Lemma also shows that $\Phi_e(A \oplus K; e)\downarrow \Leftrightarrow f(i) = 1$ where $Q_i = N_e$ we have the desired reduction. $\square$

We now wish to show that the requirements $P_{\Delta,\Gamma}$ are met. If the hypotheses of $P_{\Delta,\Gamma}$ fail then there is nothing to prove. Note, however, that in any case no subrequirement of $P_{\Delta,\Gamma}$ has an effect on any α on the true path f which is associated with any other $P_{\Delta',\Gamma'}$ (at least not after a stage after which no node to the left of α is ever accessible). Suppose, therefore, that the hypotheses of $P_{\Delta,\Gamma}$ are met. We prove by induction along f that the outcome of $\alpha \subset f$ associated with $P_{\Delta,\Gamma}$ satisfy the subrequirements of $P_{\Delta,\Gamma}$. We must also show that $\Lambda(\Delta) = Z$ is total.

Lemma 1.4. *Suppose the hypotheses of $P_{\Delta,\Gamma}$ are met and $\alpha \subset f$ is associated with $P_{\Delta,\Gamma}$.*

i) If $Q_{|\alpha|} = R_{\Delta,\Gamma,\Psi}$ then there is a stage t_0 such that at every $s \geq t_0$ either (as defined in Step 1) α is already satisfied or we satisfy α at s or α has an unrealized follower.

ii) If $Q_{|\alpha|} = S_{\Delta,\Gamma,\Psi}$ then there is a stage t_0 such that at every $s \geq t_0$ either (as defined in Step 1) α is already satisfied or we satisfy α at s or there is an n marked for α and an $x < \gamma(\alpha, n)[s]$ such that $\Psi(Z; x)$ is undefined.

Proof. We suppose that the hypotheses of $P_{\Delta,\Gamma}$ are met and proceed by induction on $\alpha \subset f$. First, note that (i) and (ii) imply that from some point on no action is taken for α except possibly in step 4 to redefine $Z(x)$ for those x which α controls. More specifically, no more numbers are used or marked or followers appointed by α; $Z(x)$ is not changed from its last value by α for any x; and no more global waits are imposed by α. All global waits for $\Gamma(\Delta, \alpha, n)\downarrow$ are eventually met by the hypotheses of $P_{\Delta,\Gamma}$. Any global wait for any n with $\Gamma(\Delta, \alpha, n)\downarrow = 1$ imposed in step 2 is permanent only if α is never removed from A (and so $\alpha \in A$). Thus, by the hypotheses of $P_{\Delta,\Gamma}$, any such condition is also eventually permanently met. Of course if $\beta <_L \alpha$, then β is accessible only finitely often and the same conditions eventually hold for β as well. Suppose, therefore, that s_0 is a stage after which no $\beta <_L \alpha$ is ever accessible but all these conditions always hold for $\beta < \alpha$.

We divide the proof into cases according to whether $\alpha \in A$ or not.

$\alpha \in A$: In this case, there is a stage $s_1 \geq s_0$ after which α is never removed from A. Let d be the number of times α is removed from A. By the hypotheses of $P_{\Delta,\Gamma}$, there is eventually a permanent current computation of $\Gamma(\Delta, \alpha, n_0) = 1$ for some $n_0 > d, s_1$, say by $s_2 \geq s_1$. Similarly, for any finite set F we must eventually have $\Gamma(\Delta, \alpha, n)\downarrow$ for every $n \in F$. Thus α must be accessible infinitely often at stages $s \geq s_2$ when all global wait conditions are satisfied.

It is now clear that if the conclusions of the Lemma do not hold then we will eventually associate some follower x with n_0 and later mark it (if $Q_{|\alpha|} = R_{\Delta,\Gamma,\Psi}$) or use and mark n_0 (if $Q_{|\alpha|} = S_{\Delta,\Gamma,\Psi}$). In either case, we would remove α from A for the desired contradiction.

$\alpha \notin A$: When α is accessible after s_0 and not in A and all global wait conditions are met, we can be prevented from putting α into A only by satisfying α. Thus we may assume that we put α into A and remove it infinitely often. By the hypotheses of $P_{\Delta,\Gamma}$, there must be an n_0 such that $\Gamma(\Delta, \alpha, n) = 0$ for every $n \geq n_0$. Consider now the ${n_0}^{\text{th}}$ time we put α into A, say at stage $s_1 \geq s_0$. We impose a global wait for an $n \geq n_0$ such that $\Gamma(\Delta, \alpha, n) = 1$. Note that we can now act for α (and so define $Z(x)$ for any new x at an $s \geq s_1$ for $Q_{|\alpha|} = R_{\Delta,\Gamma,\Psi}$) only when we either have α satisfied or an $n \geq n_0$ with $\Gamma(\Delta, \alpha, n) = 1$ which we have used.

We now assume that $Q_{|\alpha|} = R_{\Delta,\Gamma,\Psi}$ and prove (i). As we remove α from A infinitely often, we must eventually appoint and then mark an $x \geq n_0$ at stages $s_3 \geq s_2 \geq s_1$, respectively. By our induction hypotheses and choice of s_0, α retains control of x and so $Z(x)$ is, for all sufficiently large α-stages s, defined by axioms of length $u(\delta, t, s)$ (plus a constant) for a fixed t. By our hypotheses in $P_{\Delta,\Gamma}$, $\lim\limits_{s\to\infty} u(\delta, t, s)$ exists and Δ is eventually constant on all the relevant uses. If α is not already satisfied, we can be prevented from satisfying α by defining $Z(x) = 1$ at such a point only by there being some axiom σ defining $Z(x) = 0$ which was put into Δ at some stage since x was appointed and is now Δ-correct. As we have noted, any such axiom contains the information needed to either satisfy α via some other x or to give a computation $\Gamma(\Delta, \alpha, n) = 1$ for some $n \geq n_0$. As this correct Δ-information is, in fact, correct, we have the desired contradiction.

Finally, we assume that $Q_{|\alpha|} = S_{\Delta,\Gamma,\Psi}$ and prove (ii). Consider the numbers x controlled by some $\delta < \alpha$ at s_1. For each such x, $Z(x)$ is defined by an axiom of length $u(\delta, t, s)$ (plus a constant) for a fixed t. By our hypotheses on $P_{\Delta,\Gamma}$, we may choose a stage $s_2 \geq s_1$ after which $u(\delta, t, s)$ is fixed as is $\Delta \restriction u(\delta, t, s)$ for each such x. As we remove α from A infinitely often, there is a stage $s_3 \geq s_2$ at which we mark some $n > n_0$. By our choice of n_0, $\Gamma(\Delta, \alpha, n) = 0$. Once $\Delta \restriction \gamma(\alpha, n)$ has reached its correct value, we can be prevented from satisfying α (if it is not yet satisfied) only by there being an axiom σ defining some $Z(x)$ inappropriately. By our assumptions, σ was put into Λ at a stage at which we had α satisfied or an $n \geq n_0$ with $\Gamma(\Delta, \alpha, n) = 1$ and this information was included in σ. Thus this information (about Δ) must be correct and we would either have α satisfied or have a correct computation $\Gamma(\Delta, \alpha, n) = 1$ for $n > n_0$ for the desired contradiction. □

Lemma 1.5. *If the hypotheses of $P_{\Delta,\Gamma}$ are met then $\Lambda(\Delta) = Z$ (for the associated functional Λ) is well defined and total.*

Proof. As we only add a new axiom $\langle \sigma, x, j \rangle$ for x to Λ when all previous ones are incorrect ($\sigma \not\subseteq \Lambda$) and any new axiom is always at least as long as any previous one, Λ is consistent. Consider now any number x. If $x \notin \omega^{[i]}$ for some i such that Q_i is an $R_{\Delta,\Lambda,\Psi}$, $Z(x)$ is set to be 0 by a trivial axiom. Suppose $x \in \omega^{[i]}$, $Q_i = R_{\Delta,\Gamma,\Psi}$ and $\alpha \subset f$ is of length i. It is clear from the construction that α eventually takes control of x if it is not controlled by a higher priority requirement. As the priority ordering is well ordered, there is a highest priority β such that β ever controls x. Suppose β first takes control of x at t. It then controls x at every stage $s \geq t$. $Z(x)$, if undefined, is redefined when α is accessible at s via an axiom of length $u(\delta, t, s)$. Our assumptions on $P_{\Delta,\Gamma}$ guarantee that $\lim\limits_{s\to\infty} u(\delta, t, s)$ exists and so $Z(x)$ is eventually defined by a fixed (necessarily Δ-correct) axiom. □

Lemma 1.6. *If the hypotheses of $P_{\Delta,\Gamma}$ are met then $Z = \Lambda(\Delta)$ meets every subrequirement Q_i of the form $R_{\Delta,\Gamma,\Psi}$ or $S_{\Delta,\Gamma,\Psi}$ and so $0 <_T Z <_T \Delta$.*

Proof. Let α be the initial segment of f of length i and let t_0 be as in the conclusions of Lemma 1.4.

i) $Q_i = R_{\Delta,\Gamma,\Psi}$: No new followers are ever appointed after t_0. If one of them, x, is unrealized then $\Psi(x) \neq 0$ but $Z(x) = 0$ as we can never change the value of $Z(x)$ once it is defined. Thus $\Psi \neq Z$ as required. Otherwise, there is a $t_1 \geq t_0$ by which all the followers are realized. At every α-stage $s \geq t_1$, α must be satisfied, i.e., $Z(x)\downarrow = 1[s]$ and $\Psi(x) = 0[s]$ for some follower x of α. As $Z(x)$ is eventually constant by Lemma 1.4, $\Psi(x) \neq Z(x)$ for one of these x's. Again $\Psi \neq Z$ as required.

ii) $Q_i = S_{\Delta,\Gamma,\Psi}$: No numbers n are marked for α after t_0. By our hypotheses on $P_{\Delta,\Gamma}$, $\gamma(\alpha, n)[s]$ is eventually constant for all n maked for α. If $\Psi(Z; x)$ is not defined for some $x < \gamma(\alpha, n)$ for a marked n, then $\Psi(Z) \neq \Delta$. Otherwise, α is satisfied at every sufficiently large α-stage by $\Psi(Z; x) \neq \Delta(x)$ for some $x < \gamma(\alpha, n)$

for one of the marked n. As $\Delta(x)$ and $\Psi(Z;x)$ eventually stabilize for each such x, there is some x such that $\Psi(Z;x)\downarrow \neq \Delta(x)$ as required. $\square$

2. Jumps of minimal degrees in downward cones.

Theorem 2.1 *Let D be any r.e. set with $D' >_T \varnothing'$. Then there exists a set M of minimal degree with $M \leq_T D$ and $\varnothing' <_T M' \leq_T D'$.*

Proof. We will construct $M \leq_T D$ by a full approximation construction along the lines of, for instance, Lerman [1983], Epstein [1975] and Yates [1970] with $M = \lim_s M_s$. At each stage s, we shall construct nested sequences of recursive trees $T_{\alpha,s} \supseteq \ldots \supseteq T_{\lambda,s}$ for certain paths α on the stage s priority tree. At stage s, if $|\sigma| \leq s$, $|\beta| \geq e$, and σ is on $T_{\beta,s}$ then σ will have a j-state for some $j \leq e$. Such a state is a string of length $j+1$ that codes guesses as to the "arena" in which σ must live. (We will be assuming that the reader has at least nodding acquaintance, if not familiarity, with the full approximation technique. In particular, the notion of state will be somewhat similar to the usual notion of e-state, in the sense that it will characterize whether the node at hand is locally e-splitting. The notion of state will also need to encode additional Π_2 behaviour, namely the behaviour of diagonalization requirements of stronger priority as they affect the trees we search to decide if splittings exist. Because of this nonstandard notion of state, in the full construction we shall replace "state" by "guess" to avoid confusion. However, in the discussion of the basic modules we will stick to state since the need for more elaborate notions only becomes apparent when we look at interaction between the requirements.) We will meet the minimality requirements below, for all partial recursive functionals Δ_e.

$$N_e \,:\, \Phi_e(M) \text{ total } \Rightarrow [(M \leq_T \Phi_e(M)) \vee\ (\varnothing \equiv_T \Phi_e(M))]$$

Additionally, we need to ensure that $\varnothing'(= K)$ cannot compute the jump of M. In fact, we build a set V which is r.e. in M, and satisfy the following requirements.

$$R_e \,:\, \Delta_e(K) \neq V.$$

Let $Q = K^D$ denote the standard enumeration of D' so that $K_s^{D_s} = Q_s = \{e : \{e\}_s^{D_s}(e)\downarrow \,\&\, e \leq s\}$. We briefly remind the reader of the manner by which one satisfies the N_e. We assume that the reader is familiar with, for instance, the Sacks [1961] construction of a minimal degree below $\mathbf{0}'$. However, our construction will be a full approximation one as follows. As with all known minimal degree constructions, in a full approximation argument, one tries to get M on either an e-splitting (partial recursive) tree or a tree with no e-splittings. For simplicity in the following discussion we will drop some of the tree notation and pretend that we are only constructing a nested sequence $T_{0,s} \subseteq \ldots T_{s,s}$ and are working to maximise e-states on $T_{e,s}$. In the perfect set version à la Spector [1956] and Shoenfield [1971], one does this using recursive total trees, and uses $\mathbf{0}''$ as an oracle to achieve this in one step. In the present construction, one can only work locally. With no permitting around, this simply corresponds to waiting till one sees extensions τ_1, τ_2 of some node τ which are e-splitting at stage s. Clearly we need τ_1 and τ_2 on tree $T_{e,s}$, τ_1 and τ_2 having the same $(e-1)$-state as τ, and $\tau_i \subseteq M_s$ for some i. If we see such τ_1 and τ_2 we can raise τ's state to the high e-state $\nu\widehat{\ }1$, assuming it was in the low e-state $\nu\widehat{\ }0$. [We reserve ν for states; α, β, and γ for nodes on the priority tree (and sometimes states); $\mu, \sigma, \tau, \lambda$ for strings; and φ and δ for uses corresponding to Φ and Δ, respectively.] So the idea is to slowly build the trees $T_{e,s}$ as subtrees of $T_{e-1,s}$ so that $\lim_s T_{e,s}$ is achieved stringwise. At each stage s, this enables us to define a string, M_s, of length s, the leftmost common path on $T_{0,s}, \dots T_{s,s}$ of length s. By the way we define e-states and by construction, $\lim_s M_s(x) = M(x)$ will exist for all x. One can argue that M has minimal degree as in Sacks [3], except that in a full approximation argument we use e-states as follows. Let $\nu\widehat{\ }i$ ($i = 0$ or 1) be the *well-resided* e-state. By well-resided, we mean that for almost all σ on $\lim_s T_{e,s}$, if σ is an initial segment of M, then σ has state $\nu\widehat{\ }i$. If this state is $\nu\widehat{\ }1$ then we simply go to some $\sigma \subset M$ on T_e such that all $\tau \supseteq \sigma$ on T_e have final e-state $\nu\widehat{\ }1$. From this parameter, as in the Sacks construction, we know that we can inductively generate M from the tree of extensions on T_e that achieve state $\nu\widehat{\ }1$ and $\Phi_e(M)$. Similarly, if $i = 0$, then there are no e-splittings on the "well-resided tree", so any computation $\Phi_{e,s}(M_s;\, x)$ with M_s of the correct e-state must agree with $\Phi_e(M)$. [Minimality will then follow by the fact that we meet the R_e and hence $M \not\equiv_T \varnothing$.]

We remark that in the present construction, the above is not quite correct, since we will construct at each stage s a path α_s (through the priority tree) that "looks correct", and M_s will lie on the trees $T_{\gamma,s}$ for $\gamma \subseteq \alpha_s$ where α_s denotes the path of length s that looks correct at stage s. As we see, this means that the trees are actually determined also by the actions of the jump requirements. The idea, however, is essentially the same.

Keeping $M \leq_T D$ in the above construction entails adding r.e. permitting. For this theorem, we are able to use simple permitting. That is, we ensure that

$$D_s[x] = D[x] \text{ implies } M_s[x] = M[x],$$
$$\text{where } E[x] = \{z \mid z \in E \;\&\; z \leq x\}.$$

Such permitting really causes no problems with the e-state machinery. Remember, we seek e-splittings τ_1, τ_2 extending τ such that $M_s \supset \tau_1$ or $M_s \supset \tau_2$. Without loss of generality, suppose $M_s \supset \tau_1$. Then all that the minimality machinery requests is that we refine the tree to either cause $M \supset \tau_1$ or $M \supset \tau_2$ should $M \supset \tau$. The point is that we can ensure this refinement while keeping $M_s \supset \tau_1$. It follows that if $\Phi_e(M)$ is total then either $\Phi_e(M) \equiv_T M$ or $\Phi_e(M) \equiv_T \varnothing$.

What becomes more difficult is to ensure that M is nonlow by meeting the R_e. Now we are only able to diagonalize against an "α-correct" version of V when permitted by D.

Thus, we now turn to the satisfaction of the jump requirements, keeping $M' >_T \varnothing'$. To meet these requirements, the reader should recall that the plan is to build a set V which is Σ_1^M making sure that $\Delta_e(K) \neq V$. Let $\ell(e,s) = \max\{x : (\forall y < x)[\Delta_{e,s}(K_s; y) = V_s(y)]\}$. Of course, V is built by enumerating axioms relative to M. We meet these R_e requirements via a Sacks encoding type strategy, only here we desire to encode $Q = K^D$ into V via axioms about M. The idea is to try to encode more and more of Q into more and more of V so that if $\Delta_e(K) = V$, then K will be able to compute Q, since K will be able to ascertain if a coding location $c(i,s)$ for $i \in Q$ is final, for a contradiction.

So the idea is that when $\ell(e,s) > c(i-1,s)$ we choose a coding location $c(i,s)$ for coding whether "$i \in Q$". For simplicity we will not worry about the interaction with the state machinery, but consider only the basic module. The idea in the basic module is to pick some number $d(i,s)$ (intuitively the (lower bound on the) use of $c(i,s)$) so that the following four conditions are satisfied.

(i) There exists σ on $T_{e,s}$ with $\sigma = T_{e,s}(\tau)$, say, so that one of $T_{e,s}(\tau\hat{\ }0)$ or $T_{e,s}(\tau\hat{\ }1)$ is an initial segment of M_s. Without loss of generality, assume this to be $T_{e,s}(\tau\hat{\ }0)$.

(ii) $|\sigma| = d(i,s)$.

(iii) $i \in K_s^{D_s} = Q_s$ and $u(\{i\}_s^{D_s}(i)) < d(i,s)$.

(iv) $d(i,s) > t$ where t is the greatest stage, if any where $d(i,t)$ was previously defined.

If such a number exists, we declare that $d(i,s)$ is the *axiom location* for "$c(i,s) \in V$ at s". We enumerate an axiom saying that

$$\text{if } T_{e,s}(\tau\hat{\ }0) \subset M \text{ then } c(i,s) \in V.$$

Also R_e will assert control of the construction in the sense that it will constrain M to travel through $T_{e,s}(\tau\hat{\ }0)$ while the conditions above are maintained.

The key idea is that if i *leaves* Q (i.e. $i \in Q_s - Q_{s_1}$ for some least $s_1 > s$) it must be that $D_s[u(\{i\}_s^{D_s}(i))] \neq D_{s_1}[u(\{i\}_s^{D_s}(i))]$. Hence, in particular, D permits below $d(i,s)$ by (iii) above. Therefore by (i) and (ii), we can cause M to now extend $T_{e,s}(\tau\hat{\ }1)$ [instead of $T_{e,s}(\tau\hat{\ }0)$], since we are D-permitted to do so. In such a case we will abandon $T_{e,s}(\tau\hat{\ }0)$ *forever.* [This abandonment makes it necessary to ensure that some strings are set aside for other lower priority requirements, but this causes no special problems. See the remarks below.]

Now $d(i,s_1)$ becomes undefined (and hence the parameter $d(i,-)$ will need to be redefined to a new number) and $c(i,s) = c(i,s_1)$ leaves V_{s_1}. The next axiom location for $c(i,s)$, if any, will be chosen so that the relevant analogues of (i)-(iv) all hold. Note that if the cycle recurs infinitely often then $c(i,s) \notin V$ and $i \notin Q$.

The only problem that occurs with the cycle for i outlined above is the following. The cycle could cause unbounded constraint on M because $\Delta_e(K)$ may not be total, but could have unbounded use for some $c = c(i,s)$. Imagine $\delta_{e,s}(c) \to \infty$. As with the density theorem for the r.e. degrees, we might define coding locations $c(i',s)$ for all $i' > i$ and eventually this might cause M to be recursive, although $\Delta_e(K;c)\uparrow$, since $\limsup \ell(e,s) = \infty$. To overcome this familiar problem, when we pick a coding location $c(i,s)$ for i at stage s if the $\Delta_e(K_s;c)\downarrow$ computation is later seen to have incorrect K-use we cancel $c(i',s)$ for all $i' > i$ and pick $c(i',t)$ anew later (where $c(i',t) > s$) (we call this *kicking*). For all such $c(i',s)$, we also declare $d(i',s)$ to be undefined; we do not cancel the corresponding $T_{e,s}(\tau\hat{\ }0)$ or $T_{e,s}(\tau\hat{\ }1)$ but release them to the construction.

The conclusion that $\Delta_e(K) \neq V$ remains correct since using $\delta_{e,s}$ and K_s, K can still decide if a coding location is final.

Finally, in the full construction, we will need to deal with various versions of the R_e and N_e above working in a "tree of strategies" type of setting. Coding locations with the wrong guesses will need to be initialized and kicked as above. An R_e requirement at node α which is guessing that ν is the well-resided "e" guess (note "guess" not "state") will need to wait till it sees σ's and τ's of guess ν before it enumerates an axiom involving an axiom location $d(\alpha, i, s)$ for the coding location $c(\alpha, i, s)$. The reader should note that the trickiest part of the construction is to get the correct environment within which each R_e operates. Here, not only do we need to guess the $\liminf$ of the length of agreement associated with R_{e-1}, but the exact behavior of the various markers associated with this $\liminf$. We now turn to the formal details.

Remarks. As we mentioned above, the reader should note that a single $c(\alpha, i, s)$ has the potential of killing all of the tree T_α save for one branch (making M recursive) if we are not careful. If $i \notin Q$ but i is in Q_s infinitely often, then the R_e cycle could be repeated infinitely often for the sake of i. As above the action of the R_e is to kill $T_\alpha(\sigma^\frown j)$ and if we then killed one extension of $T_\alpha(\sigma^\frown(1-j))$ and so forth, one could kill all but one path on the tree. This is fine from R_e's point of view, but has the potential for making M recursive. To ensure that this problem is avoided we shall use various control devices to guarantee that there will be *enough* available branches for the relevant requirements when we play the appropriate outcome of R_e. This is a little messy but is the point of the control functions $\mathrm{count}(\alpha, s)$ and $\mathrm{maxcount}(\alpha, s)$, and the *freezing* machinery. The idea is that suppose we wish to play an outcome o of α, as its guess has just looked correct. At this stage we acknowledge that we want a supply of strings available to the nodes guessing $\alpha^\frown o$ when we pass $\alpha^\frown o$. We will not allow α, or any guesses extending $\alpha^\frown o$, to assign any strings until we see $\mathrm{count}(\alpha^\frown o, s)$ exceed a previous bound $\mathrm{maxcount}(\alpha^\frown o, s)$, or we unfreeze the node because we see many strings of the appropriate guess. We increment $\mathrm{count}(\alpha^\frown o, s)$ each time we visit α and would like to play $\alpha^\frown o$. In this way, when we actually get to play $\alpha^\frown o$ we have many strings to assign to nodes guessing $\alpha^\frown o$.

The priority tree. We generate the priority tree PT by induction first on length and then on the lexicographic ordering generated by the ordering $<$ that we define on outcomes. We also denote this lexicographic ordering on nodes by $<$. It is the usual priority ordering. The left-to-right ordering $\alpha <_L \beta$ is defined as usual by $\alpha < \beta$ and $\alpha \nsubseteq \beta$. On any path ρ a requirement is assigned to a node $\sigma \subset \rho$ as follows. Assign N_0 to λ, the empty string. We write $e(\lambda) = 0$ and $\mathrm{req}(\lambda) = N_0$. λ will have outcomes 0 and 1, meaning that $\lambda^\frown 0 = 0$ and $\lambda^\frown 1 = 1$ are on PT. The idea here is that the outcome 1 corresponds to M lying upon a 0-splitting tree and we set $1 < 0$. To each of λ's outcomes we now assign a version of R_0 so that $\mathrm{req}(0) = \mathrm{req}(1) = R_0$, and $e(0) = e(1) = 0$.

Each version of R_0 will sit atop an infinite *tree* of outcomes of R_0, the R_0 *tree*, coding the precise behavior of the coding locations, axiom locations and K-uses. At the top level will be the node σ with the $c(0)$-outcomes of R_0. Namely, σ will have outcomes (in $<$ order)

$$\langle 0, u\rangle,\ \langle 0, \infty\rangle,\ \langle 0, d\rangle,\ \langle 0, f\rangle.$$

The intended meaning of these outcomes is as follows:

$\langle 0, u\rangle$: $c(\sigma, 0)$ $(= \lim_s c(\sigma, 0, s))$ has unbounded $\Delta_0(K)$-use ("u" for "unbounded").

$\langle 0, \infty\rangle$: This corresponds to no win via $c(\sigma, 0)$, but we change $d(\sigma, 0, s)$ infinitely often. In essence, this entails $\Delta_0(K; c(\sigma, 0)) = 0$ and cycling through $\langle 0, d\rangle$ infinitely often.

$\langle 0, d\rangle$: $c(\sigma, 0)$ witnesses a win by a disagreement ("d" for "disagreement").
$\langle 0, f\rangle$: No win via $c(\sigma, 0)$ and we only change $d(\sigma, 0, s)$ finitely often ("f" for "finite").

Remark. The reader should note that the outcomes $\langle 0, \infty\rangle$ and $\langle 0, f\rangle$ do not code a win for σ but do code the behavior of σ's action with respect to $c(\sigma, 0)$. The idea is that we will continue to try to satisfy R_0 at each of these outcomes generating an R_0 tree as described below.

In general to define the R_0 tree below σ suppose we have defined τ to be in this tree. If τ is of the form $\mu\widehat{\ }\langle i, u\rangle$ or $\mu\widehat{\ }\langle i, d\rangle$, then τ represents a win for R_0 and we would assign N_1 to τ. If τ is of the form $\mu\widehat{\ }\langle i, \infty\rangle$ or $\mu\widehat{\ }\langle i, f\rangle$, then we again assign R_0 to τ and give it outcomes $\langle i+1, u\rangle, \dots, \langle i+1, f\rangle$. (See Diagram 1.)

For the full priority tree we assign N_e and R_e in order as above with e in place of 0 for the highest priority requirement not yet (completely) assigned. Note that the presence of R_e trees means that there exist paths P on PT whose nodes are only assigned to finitely many requirements. For such a path, almost all of the nodes will be assigned to a fixed R_e and will be of the form $\mu\widehat{\ }\langle i, \infty\rangle$ or $\mu\widehat{\ }\langle i, f\rangle$. We will need to argue that none of these paths is the true path of the construction, and hence all requirements get their chance to be met.

Parameters and Terminology for the Construction.

$c(\alpha, i, s)$: The coding marker associated with " $i \in D'$ " at guess α at stage s. $c(\alpha, i, s)$ is targeted for V. We attempt to code D' into V via these markers.

$d(\alpha, i, s)$: The current axiom location for "$c(\alpha, i, s)$ in V at stage s."

$\text{count}(\alpha, s)$: A control function at guess α used to ensure that when we play an N_e-node α then there are many unused strings available for guesses below α.

$\text{maxcount}(\alpha, s)$: The reference control function that $\text{count}(\alpha, s)$ must exceed before we can play α.

α-freezing : While a node α is waiting to see enough strings to make it seem reasonable to play outcome o then we will say that the node is *frozen with outcome* o.

$\text{req}(\alpha)$: The requirement assigned to α.

$e(\alpha)$: The index of the requirement $\text{req}(\alpha)$.

$j(\alpha)$: The j for which α is used to code "$c(\alpha, j, s) \in V$."

$\ell(e(\alpha), s) = \max\{x : \forall y < x(\Delta_{e,s}(K_s; y) = V_s(y)))\}$.

The most complex requirements are the higher level Sacks coding requirements, the R_e. The reader should think of their action as that of an automaton.

Construction, stage $s+1$:

In substages we shall generate a guess (i.e. a node on the tree) α_{s+1} that looks correct, and a collection of trees $T_{\gamma,s+1}$(for $\gamma \subseteq \alpha_{s+1}$ and $\text{req}(\gamma)$ of the form N_e). For convenience, let $T_{\lambda,0} = 2^{<\omega}$. Initially, let all μ on $T_{\gamma,0}$ have e-guess λ. We proceed in substages $t \leq s+1$. We will append a superscript t to a parameter to indicate its guess at the end of substage t. (At substage 0, the initial value of a parameter will be its value at the end of stage s). We also suppose that we are told that $s+1$ is an α-stage ($\alpha = \alpha^t_{s+1}$) at the end of each substage t ($\alpha = \lambda$ initially).

Our action will be to determine how to modify trees, guesses, etc., and to define the outcome of α^t_{s+1}. Our actions will be determined by s, t, and $\text{req}(\alpha)$. We will assume that if $\mathrm{req}(\alpha) = R_e$, we are at the top of an R_e tree. We often use the phrase "initialize α." This follows standard usage, and means cancelling α's followers, etc. Furthermore, if $\beta <_L \alpha$ initializes α (β is visited at $s+1$, say) then we let $T_{\alpha,s+1}$ be the full subtree of $T_{\beta,s+1}$ extending M_{s+1} above $T_{\beta,s+1}(\tau)$) with $|\tau| = s+1$ and $M_{s+1} \supset T_{\beta,s+1}$, with all guesses becoming initialized. Finally setting a value for α_{s+1} ends a stage.

Convention. We will also adopt the following convention that saves on terminology. During the construction we can visit nodes ν in one of two ways. One way is to visit them at substage t of stage s such that for all $\nu' \subseteq \nu$ we also visited ν' at the same stage. The other possibility is that we did not visit some $\nu' \subseteq \nu$ during stage s but jumped directly to ν because although ν' was not accessible, there was a permitted action at ν. In the latter case, we adopt the convention that visiting ν either ends the stage or we visit some $\widehat{\nu} \supset \nu$ via a permitted action. As we shall see when we act in this way we do not appoint markers etc. but merely modify trees.

Definition 2.2. We say that α (on PT) has a *permitted action* at substage t of stage $s+1$ if, for some j,
(i) $c = c^t(\alpha, j, s+1)$ is defined,
(ii) $c \in V^t_{\alpha,s+1}$, and
(iii) $j \notin Q_{s+1}$ (i.e. $\{j\}^D(j)[s+1]\uparrow$).

Substage t: Let α be the node eligible to act at substage t.

Case 1: $\mathrm{req}(\alpha)$ is N_e.

Subcase 1: There exist τ, τ_0, τ_1 such that τ, τ_0, and τ_1 are on $T^t_{\alpha,s+1}$ with $\tau = T^t_{\alpha,s+1}(\eta)$, $\tau_1 = T^t_{\alpha,s+1}(\eta_1)$, $\tau_2 = T^t_{\alpha,s+1}(\eta_2)$, and

(i) τ_0 and τ_1 e-split τ,
(ii) $\tau_0 <_L \tau_1$,
(iii) $\tau_0 \subseteq M_s$ or $\tau_1 \subseteq M_s$,
(iv) τ, τ_0, τ_1 had e-guess $\alpha\widehat{\ }0$ at the beginning of substage t,
(v) $|\eta_0|, |\eta_1| >$ maxcount$(\alpha\widehat{\ }1, s)$ (the last stage at which α genuinely had outcome 1), and
(vi) τ is the shortest such string.

Action: For $\rho \not\supset \eta$, let $T^{t+1}_{\alpha,s+1}(\rho) = T^t_{\alpha,s+1}(\rho)$. For $\rho \supset \eta$ and $i < 2$, define $T^{t+1}_{\alpha,s+1}$ via

$$T^{t+1}_{\alpha,s+1}(\eta\widehat{\ }i\widehat{\ }\sigma) = T^t_{\alpha,s+1}(\eta_i\widehat{\ }\sigma).$$

Raise the e-guess of τ to $\alpha\widehat{\ }1$. All else remains the same.

Set count$(\alpha\widehat{\ }1, s+1) =$ count$(\alpha\widehat{\ }1, s) + 1$. There are three subcases.

Subcase 1a: count$(\alpha\widehat{\ }1, s+1) <$ maxcount$(\alpha\widehat{\ }1, s)$ and there exists a $\beta \supseteq \alpha\widehat{\ }1$ with a permitted action.

Action: Set $\alpha^{t+1}_{s+1} = \beta$ and declare $s+1$ to be an $\alpha\widehat{\ }\gamma$-stage for all $\alpha\widehat{\ }\gamma \subseteq \beta$. [For all γ with $\alpha\widehat{\ }1 \subseteq \alpha\widehat{\ }\gamma \subsetneq \beta$, however, $s+1$ not a *genuine* $\alpha\widehat{\ }\gamma$-stage. A stage is a genuine η-stage if we actually visit η at one of its substages.]

Subcase 1b: count$(\alpha\widehat{\ }1, s+1) <$ maxcount$(\alpha\widehat{\ }1, s)$ but Subcase 1a fails to apply.

Action: Set $\alpha_{s+1} = \alpha\widehat{\ }0$ and declare $s+1$ to be an $\alpha\widehat{\ }0$-stage. [This ends the stage. Recall that a low count indicates we have not yet seen enough potential strings in the good guess.]

Subcase 1c: $\mathrm{count}(\alpha\widehat{\ }1, s+1) \geq \ \mathrm{maxcount}(\alpha\widehat{\ }1, s)$.

Action: Set $\mathrm{maxcount}(\alpha\widehat{\ }1, s+1) = s+1$ and reset $\mathrm{count}(\alpha\widehat{\ }1, s+1) = 0$. Declare $s+1$ to be an $\alpha\widehat{\ }1$-stage and set $\alpha^{t+1}_{s+1} = \alpha\widehat{\ }1$.

Subcase 2: Subcase 1 does not hold, but there exists a permitted action at some $\beta \supseteq \alpha\widehat{\ }1$.

Action: For the highest priority such β (that is, the lexicographically least) let α^{t+1}_{s+1} be β and declare $s+1$ to be an $\alpha\widehat{\ }\gamma$-stage for all $\alpha\widehat{\ }\gamma \subseteq \beta$. [For all γ with $\alpha\widehat{\ }1 \subseteq \alpha\widehat{\ }\gamma \subsetneq \beta$, however, $s+1$ is not a *genuine* $\alpha\widehat{\ }\gamma$-stage]. Otherwise nothing changes.

Subcase 3: Otherwise.

Action: Nothing changes. Set $\alpha^{t+1}_{s+1} = \alpha\widehat{\ }0$. Declare $s+1$ to be an $\alpha\widehat{\ }0$-stage.

Case 2: $\mathrm{req}(\alpha)$ is R_e. We begin with the case that α is the top of an R_e-tree. We continue to consider R_e until we play either an outcome $\langle j, d\rangle$ or $\langle j, u\rangle$ for some j. We consider the j in order, beginning with $j = 0$. Suppose $j = 0$ or we have already considered $j-1$ and are considering j. We can assume that we have inductively generated a string $\alpha\widehat{\ }\gamma(j)$ so that $\alpha\widehat{\ }\gamma(j) \subseteq \alpha_{s+1}$ and for all $j' < j$, there exist η, k with $\alpha\widehat{\ }\eta\widehat{\ }\langle j', k\rangle \subseteq \alpha\widehat{\ }\gamma(j)$ and $k \in \{\infty, f\}$. [For $j = 0; \gamma(j) = \lambda$.] Finally, we let $T^t_{\eta,s+1} = T^t_{\alpha,s+1}$ for all η extending α which are devoted to R_e below α, i.e. they all use the same tree.

Pick the first subcase that applies and perform the action indicated. Here let s_0 be the last genuine $\alpha\widehat{\ }\gamma(j)$-stage at (the beginning of) which $\alpha\widehat{\ }\gamma(j)$ was not frozen.

Subcase 1: $c = c^t(\alpha, j, s+1)$ is undefined.

Action: Pick a large fresh number $c = c^{t+1}(\alpha, j, s+1)$ and set $\alpha_{s+1} = \alpha^{t+1}_{s+1} = \alpha\widehat{\ }\gamma(j)$.

Subcase 2: α has a permitted action at substage t of stage $s+1$.

Action: Let $d = |T^t_{\alpha,s+1}(\tau\widehat{\ }i)|$ be the use of "$c \in V^t_{\alpha,s+1}$" (where $T^t_{\alpha,s+1}(\tau\widehat{\ }i) \subset M^t_{s+1}$); abandon the part of $T^t_{\alpha,s+1}$ above $T^t_{\alpha,s+1}(\tau\widehat{\ }i)$ by setting

$$T^{t+1}_{\alpha,s+1}(\eta) = \begin{cases} T^t_{\alpha,s+1}(\eta), & \text{if } \eta \not\supseteq \tau \\ T^t_{\alpha,s+1}(\tau\widehat{\ }(1-i)\widehat{\ }\eta_0), & \text{if } \eta \supseteq \tau, \text{ say } \eta = \tau\widehat{\ }\eta_0 \end{cases}$$

(and so $c \notin V^{t+1}_{\alpha,s+1}$); let M^{t+1}_{s+1} be a path through $T^{t+1}_{\alpha,s+1}$ extending $T^{t+1}_{\alpha,s+1}(\tau\widehat{\ }(1-i))$; declare $\alpha\widehat{\ }\gamma(j)$ frozen with outcome u (if $\alpha\widehat{\ }\gamma(j)$ was already frozen with outcome u or if $\Delta^K_e \restriction (c+1)$ was not defined at any time since stage s_0) or with outcome ∞ (otherwise); and set $\alpha^{t+1}_{s+1} = \alpha\widehat{\ }\gamma(j)\widehat{\ }\langle j, u\rangle$ or $\alpha\widehat{\ }\gamma(j)\widehat{\ }\langle j, \infty\rangle$, respectively.

Subcase 3: $\alpha\widehat{\ }\gamma(j)$ is frozen (with outcome o since stage s_0, say).

Action: Check if there are at least s_0 many strings τ such that

(i) τ is on $T^t_{\alpha,s+1}$, say $\tau = T^t_{\alpha,s+1}(\nu)$,

(ii) $\tau \subset M^t_{s+1}$,

(iii) the $e(\eta)$-guess of $T^t_{\alpha,s+1}(\nu\widehat{\ }i)$ is $\alpha(e(\eta))$ for all N-strategies $\eta \subset \alpha$ and all $i < 2$, and

(iv) $|(T^t_{\alpha,s+1})^{-1}(\tau)| > s_0$.

If so then declare $\alpha^\frown\gamma(j)$ no longer frozen, and let $\alpha^\frown\gamma(j)$ have outcome o. If furthermore
(v) $\alpha^\frown\gamma(j)$ was frozen with outcome ∞, and
(vi) $c \notin V^t_{\alpha,s+1}$ and $j \in Q_{s+1}$ (i.e. $\{j\}^D(j)[s+1]\downarrow$),
then enumerate c into $V^{t+1}_{\alpha,s+1}$ with use $T^t_{\alpha,s+1}(\nu^\frown i)$ where τ is the longest string satisfying (i)-(iv) not used by a strategy $\subset \alpha^\frown\gamma(j)$ for V-enumeration and $T^t_{\alpha,s+1}(\nu^\frown i) \subset M^t_{s+1}$.

Otherwise, i.e. if (i)-(iv) fail, $\alpha^\frown\gamma(j)$ remains frozen with outcome o. In either case, we set $\alpha^{t+1}_{s+1} = \alpha^\frown\gamma(j)^\frown\langle j, o\rangle$.

Subcase 4: $\Delta^K_e \restriction (c+1)$ has not always been defined since stage s_0.
Action: Declare $\alpha^\frown\gamma(j)$ frozen with outcome u and set $\alpha^{t+1}_{s+1} = \alpha^\frown\gamma(j)^\frown\langle j, u\rangle$.

Subcase 5: $Q(j)$ has changed since stage s_0, i.e. there are stages $s_1, s_2 \in [s_0, s+1]$ such that exactly one of $\{j\}^D(j)[s_1]\downarrow$ and $\{j\}^D(j)[s_2]\downarrow$ holds.
Action: Declare $\alpha^\frown\gamma(j)$ frozen with outcome ∞ and set $\alpha^{t+1}_{s+1} = \alpha^\frown\gamma(j)^\frown\langle j, \infty\rangle$.

Subcase 6: $j \in Q_{s+1}$ and $c \notin V^t_{\alpha,s+1}$.
Action: Declare $\alpha^\frown\gamma(j)$ frozen with outcome ∞ and set $\alpha^{t+1}_{s+1} = \alpha^\frown\gamma(j)^\frown\langle j, \infty\rangle$.

Subcase 7: $\Delta^K_e \restriction (c+1) = V_\alpha \restriction (c+1)$ has not always held since stage s_0.
Action: Declare $\alpha^\frown\gamma(j)$ frozen with outcome d and set $\alpha^{t+1}_{s+1} = \alpha^\frown\gamma(j)^\frown\langle j, d\rangle$.

Subcase 8: Otherwise.
Action: Let $\alpha^\frown\gamma(j)$ have outcome f and set $\alpha^{t+1}_{s+1} = \alpha^\frown\gamma(j)^\frown\langle j, f\rangle$.

This completes the action of $\alpha^\frown\gamma(j)$. We now initialize all strategies $\xi >_L \alpha^{t+1}_{s+1}$ and check if
(i) $\alpha^\frown\gamma(j)$ is now frozen and there is a strategy $\beta <_L \alpha^{t+1}_{s+1}$ or $\supseteq \alpha^{t+1}_{s+1}$ with a permitted action (as defined in Subcase 2 above), or
(ii) there is a strategy $\beta <_L \alpha^{t+1}_{s+1}$ with a permitted action.

If so then let the highest-priority such β act next. Otherwise, let α^{t+1}_{s+1} act at the next substage (if $\alpha^\frown\gamma(j)$ is not frozen now and Subcase 1 above did not apply) or end the stage (otherwise).

End of construction.

Verification.

The reader should note that $M \leq_T D$ because we always maintain the simple permitting invariant : $M_s[x] = M[x]$ if $D_s[x] = D[x]$. [While we might refine the underlying trees, we only change M due to a permitted action.] Let TP denote the true path.

We need to argue that for all $\alpha \subseteq TP$, the following hold:

(i) $\lim_s T_{\alpha,s} = T_\alpha$ exists stringwise.
(ii) Let $\alpha = \alpha^{-\frown} a$. If $\mathrm{req}(\alpha^-) = N_e$, then for almost all σ on T_α with $\sigma \subseteq M$ the $e(\alpha^-)$-guess of σ agrees with α.
(iii) If $\alpha \subseteq TP$ then there are infinitely many α-stages and α is only initialized finitely often.
(iv) If $\mathrm{req}(\alpha) = R_e$ then there are at most finitely many α' on TP with $\mathrm{req}(\alpha') = R_e$ and R_e is met by some such α'. Furthermore, if $\mathrm{req}(\alpha) = \mathrm{req}(\alpha')$ then for all η with $\alpha \subseteq \eta \subseteq \alpha'$, $\mathrm{req}(\eta) = R_e$.

(v) Let α be the shortest string on TP with $\mathrm{req}(\alpha) = R_e$ and let τ be any such string. Let $\tau\widehat{\ }a \subset TP$. Then a is of the form $\langle j, \ \rangle$, and $\lim_s c(\alpha, j, s) = c(\alpha, j)$ exists. Furthermore the following hold:

(va) If $a = \langle j, u\rangle$ then the K-use of $\Delta_e(K; c(\alpha, j))$ is unbounded.

(vb) If $a = \langle j, \infty\rangle$ then $\Delta_e(K; c(\alpha, j)) = 0$ and $j \notin Q, c(\alpha, j) \notin V$, and $d(\alpha, j, s)$ tends to infinity as s does.

(vc) If $a = \langle j, d\rangle$ then $\Delta_e(K; c(\alpha, j)) \neq V(c(j))$ and $d(\alpha, j, s)$ is only reset finitely often.

(vd) If $a = \langle j, f\rangle$, then $\Delta_e(K; c(\alpha, j))$ is defined, equals $V(c(j))$ and $d(\alpha, j, s)$ is reset only finitely often.

These are all fairly straightforward and are verified by simultaneous induction. Let $\alpha \subseteq TP$ and let s_0 be a stage such that we are never again to the left of α, and α is not initialized after s_0.

We begin with (i). If $\mathrm{req}(\alpha)$ is R_e, then α is devoted to some subrequirement attempting to produce a disagreement at some j in the R_e tree below some σ. Note that all the α' in this tree work on the same T_σ. Now α only modifies trees in Subcase 2, where it deletes nodes, and redefines the tree. But whenever we do this we pick ν at the end of Subcase 3 much longer the next time. From this and the induction hypothesis it follows that $\lim_s T_{\alpha,s} = \lim_s T_{\sigma,s}$ exists stringwise. In the case that $\mathrm{req}(\alpha) = N_e$, we see that (i) follows immediately for α since we were permitting, and only raise guesses.

To see that (ii) holds, after stage s_0, we know that α^- and α will not be initialized, and that $\mathrm{req}(\alpha^-) = N_e$. If $\alpha = \alpha^-\widehat{\ }1$, then we only get to play α when we see a new splitting (Case 1, Subcase 1), or there is a permitted action at some $\beta \supseteq \alpha$. In either case, the only places we will change M_s will be on trees T_γ for $\gamma \supseteq \alpha$, and by induction, these will only involve strings of e guess α. Note that the same can be said of the case $\alpha = \alpha^-\widehat{\ }0$ since we will not be to the left of α again.

Turning to (iii), we can suppose that there are infinitely many α^--stages and all of the above holds for α^-. If $\mathrm{req}(\alpha^-) = N_e$, then we know $\alpha = \alpha^-\widehat{\ }0$ or $\alpha = \alpha^-\widehat{\ }1$. In the former case, after some stage we will set a new count which is never exceeded. (Otherwise, we would move left of α.) This means that $T_{\alpha,s}$ is never again reset and all the nodes on it are of guess $\subseteq \alpha^-\widehat{\ }0$. Since (iii) holds for nodes $\subseteq \alpha^-$, it follows that α^- is given arbitrarily large numbers of strings of guess α^-. These strings constitute an infinite collection with guess $\alpha^-\widehat{\ }0 = \alpha$. The case $\alpha = \alpha^-\widehat{\ }1$, is similar, only there we know that infinitely often we see further strings with guess α^-. We only need to argue that infinitely often $\mathrm{count}(\alpha^-, s+1)$ is exceeded.

We remark that the same argument works for $\mathrm{req}(\alpha^-) = R_e$ via the freezing machinery in place of the count machinery. Thus a node is frozen and waits in Subcase 3. The result now follows from (ii) and the induction hypothesis applied to α^-.

To establish (iv) and (v) we argue exactly as in the basic module. The trees T_α for $\alpha \subseteq TP$ are not initialized after a certain stage s_0. Let α be the top of an R_e tree with $\alpha \subseteq TP$. All its nodes $\beta \supset \alpha$ are devoted to R_e and T_α. Now the key point is that for nodes in the R_e-tree of the form $\alpha\widehat{\ }\gamma(j) = \alpha^j$, if $d = d(j, \alpha, s)$ is defined, then whenever j leaves Q at stage $s+1$ we will be able to cancel d via Subcase 2 as there will be a permitted action at α^j and hence s will either be a genuine α^j-stage, or s will be a β-stage for some β to the left of, or below α^j. In the latter case we will have picked d after we visited β and hence β's action will *also* move M_s from extending the string corresponding to d at α^j. The remaining details are to do a routine case by case analysis of the subcases to show that our construction mirrors the basic module, with the additional freezing delay put in to ensure that $\beta \supset \alpha$ have enough strings to work with. These details are by and large straightforward, and should probably be apparent now to the reader, but we supply some as examples. First note that (va) holds since we only play the outcome $\langle j, u\rangle$ when the K use of $\Delta_e(K; c(\alpha, j))$ has been seen to increase. Since $\tau\widehat{\ }\langle j, u\rangle \subset TP$, we know that $c(\alpha, j)$ reaches a limit and hence the fact that the use increases infinitely often implies that for the final $c(\alpha, j)$ the use is unbounded. Similarly for (vb), we must have that $\Delta_e(K; c(\alpha, j))$ equals 0. As with the basic module, each time we unfreeze τ we will get to assign $d(\alpha, j, s)$ corresponding to the same $c(\alpha, j)$, and after this j will later leave Q, and hence (vb) follows since then the Δ_e use has finite limit but $\{j\}^D_s(j)\uparrow$ by divergence. The other two cases are similar. Finally, to see that (iv) holds, suppose that it fails. Then below α the only nodes on the true path are ones of the form $\tau\widehat{\ }a$ with $e(\tau) = e = e(\alpha)$, and $a \in \{\infty, f\}$. We claim that K can compute Q. To see this once α will never again be initialized, any $c(\alpha, j, s)$ once defined will be fixed unless the K-use of $c(\alpha, i, s)$ changes for some $i < j$. We know that all such uses reach a limit so for each j, some incarnation of $c(\alpha, j, s)$ is eventually fixed. Furthermore, note that since K can figure out if the use at c is final, K can figure out if an incarnation of $c(\alpha, j)$ is the final one. By induction, and the argument above we see that (vb) or (vd) applies to all nodes on the true path above α. We can therefore apply the argument of the basic module, and conclude that $j \in Q$ iff $c(\alpha, j) \in V$.

The argument that M is minimal is totally routine. To see that all the N_e are met, pick ρ on TP devoted to N_e, and look at its outcomes. First, since ρ is only initialized finitely often, for almost all strings on T_ρ, ρ will be able to raise guesses at will. Because of this the standard argument will work. We leave the details to the reader. $\square$

We now immediately have the desired corollary showing that the range of the jump operator on the minimal degrees below $\mathbf{0}'$ cannot be characterized simply in terms of the jump classes of degrees r.e. in and above $\mathbf{0}'$.

Corollary 2.3. *If* $\mathbf{c} > \mathbf{0}'$ *is r.e. in* $\mathbf{0}'$, *then there is a minimal degree* $\mathbf{m} < \mathbf{0}'$ *such that* $\mathbf{0}' < \mathbf{m}' \leq \mathbf{c}$.

3. Some related results on minimal degrees.

We now sketch the proof of two related results on jumps of minimal degrees below $\mathbf{0}'$. We assume familiarity with the standard oracle construction of a minimal degree below $\mathbf{0}'$ introduced by Shoenfield [1966] as in Lerman [1983, IX.2] (whose notation we adopt below) and the diagonalization method introduced by Sasso [1974] using what Lerman [1983, V.3] calls narrow trees to construct a minimal degree $\mathbf{a} < \mathbf{0}'$ with $\mathbf{0}' < \mathbf{a}'$ as in Lerman [1983, V.3 and IX.2.11]. We also assume familiarity with the standard method of recursively approximating the answer to a Σ^B_1 question for low B originally introduced in Robinson [1971] as in Soare [1987, XI.3]. We use this method relativized to $\mathbf{0}'$.

Theorem 3.1. *If* $\mathbf{c}$ *is r.e. in and above* $\mathbf{0}'$ *and* $\mathbf{c}' = \mathbf{0}''$ *then (uniformly in the information) there is a minimal degree* $\mathbf{a} < \mathbf{0}'$ *with* $\mathbf{a}' \not\leq \mathbf{c}$.

Proof (Sketch). We replace the nonrecursiveness requirements in the oracle construction of a minimal degree $\mathbf{a} < \mathbf{0}'$ by ones to guarantee that $\Phi_i(C) \neq A'$. At each stage s of the standard construction we would have a sequence of partial recursive trees $T_{0,s}, \ldots, T_{n_s,s}$ such that $T_{i+1,s}$ is either $Ext(T_{i,s}, \sigma)$, the full subtree of $T_{i,s}$ above σ, for some σ or $Sp(T_{i,s}, i)$, the (partial recursive) i-splitting subtree of $T_{i,s}$. Instead, we now have a sequence in which $T_{2i+1,s}$ is either a full subtree of $T_{2i,s}$ or $Nar(Ext(T_{2i,s}, \sigma))$, the narrow subtree of some full subtree of $T_{2i,s}$. (The narrow subtree of T is the tree which eliminates the right hand branches above all nodes at odd levels of T. $Nar(T)(\sigma) = T(\sigma \oplus 0^n)$ where $n = lh(\sigma)$.) Moreover, $T_{2i+2,s}$ is either a full subtree of $T_{2i+1,s}$ or the i-splitting subtree of $T_{2i+1,s}$.

It is easy to see that we can recursively in (the index for) $T_{2i,s}$ calculate an x such that if A is on $T_{2i,s}$ then $x \notin A'$ iff A is on $T_{2i+1,s}$, the narrow subtree of $T_{2i,s}$. If, at stage s of our construction, we see that $\Phi_i(C; x)[s] = 0$ then we would like to let $T_{2i+1,s+1}$ be some full subtree of $T_{2i,s}$ which forces A off $T_{2i+1,s}$ such as $Ext(T_{2i+1,s}, 01)$. The problem is that, as the construction is recursive in $\mathbf{0}'$ and C is only r.e. in $\mathbf{0}'$, this computation may prove false. We use the low oracle approximation procedure to prevent us from acting infinitely often for this requirement.

At stage s we first redefine the trees $T_{0,s}, \ldots, T_{n_s,s}$ as in the standard construction (to attempt to satisfy the minimality condition). We then see if there is a $2i < n_s$ for which we do not think we have satisfied the requirement $\Phi_i(C) \neq A'$; $\Phi_i(C; x) = 0[s]$ where x is calculated from $T_{2i,s}$ as above; and our oracle approximation procedure says that there is a stage with these properties at which the C computation is actually correct. (If the approximation says "no" we speed up our enumeration of C and the approximation until either we no longer have $\Phi_i(C; x) = 0$ or the approximation says "yes.") If so, we let $T_{2i+1,s+1}$ be a full subtree of $T_{2i,s}$ ($= T_{2i,s+1}$) which forces A off $T_{2i+1,s}$. (As usual, $n_{s+1} = 2i + 1$.) If not, we make no additional changes in the trees to get the other $T_{i,s+1}$.

The only new element of the verification is the analysis of the effects of moving off the narrow subtrees. The lowness of C over $\mathbf{0}'$ (and the recursion theorem relative to K) guarantees that our K-recursive approximation is correct and so we act only finitely often to satisfy the requirement $\Phi_i(C) \neq A'$ and we eventually C-correctly satisfy it. As this action is finite, the argument that the trees $T_{i,s}$ are eventually constant and the minimality requirements are satisfied proceeds as usual. □

A nonuniform version of the above result also follows from the following.

Theorem 3.2. *There are minimal degrees* $\mathbf{a}_0, \mathbf{a}_1 < \mathbf{0}'$ *such that* $\mathbf{a}_0' \cup \mathbf{a}_1' = \mathbf{0}''$.

Proof (Sketch). Simultaneously build two sets A_0, A_1 of minimal degree as above but use the narrow subtrees in each construction to code the final result of the other in an interleaved way as in Simpson [1975]. Suppose the sequences of trees for A_0 and A_1 at stage s are $\langle T^0_{0,s}, \ldots, T^0_{n,s} \rangle$ and $\langle T^1_{0,s}, \ldots, T^1_{n_s,s} \rangle$, respectively, and $\alpha_{j,s} = T^j_i(\sigma_{i,j}, s)$ for $j = 0, 1$. We will also use the narrow subtrees to code $\varnothing''$ into $A_0' \vee A_1'$ in place of diagonalization. At each stage s of our construction we will have, for $j = 0, 1$, a sequence of (partial recursive) trees $\langle T^j_{0,s}, \ldots, T^j_{n_s,s} \rangle$ and an initial segment $\alpha_{j,s}$ of A_j with strings $\sigma_{i,j,s}$, $i \leq n_s$, such that $T^j_{i,s}(\sigma_{i,j,s}) = \alpha_{j,s}$. In particular, $\sigma_{n_s,j,s} = \varnothing$. $T^j_{0,s} = \text{id}$ for every s while, for $k \geq 0$, $T^j_{2k+j+1,s}$ will be either the k–splitting subtree of $T^j_{2k+j,s}$ or a full subtree of $T^j_{2k+j,s}$. Alternating with these trees, we will have ones $T^j_{2k+j+2,s}$ $(k \geq 0)$ which will be narrow subtrees of some full subtree of $T^j_{2k+j+1,s}$.

At stage s, we first find the least k such that T^j_{2k+j+1} is not defined at $\alpha_{2k+j+1,s} * 01$, $(j = 0, 1)$. If there is an $i < k$ enumerated in $\varnothing''$ at s, we record this fact by forcing A_1 off the narrow subtree of $T^1_{2i,s}$. We let $n_{s+1} = 2i+1$ and $T^1_{2i+1,s+1} = Nar(Ext(T^1_{2i}, \alpha_{2i,1,s} * 01))$. No other trees are changed.

If no $i < k$ is enumerated in $\varnothing''$ at stage s and $2k+j+1 \leq n_s$, we let $n_{s+1} = 2k+j+1$ and set $T^j_{n_{s+1},s+1} = Ext(T^j_{2k+j,s}, \tau)$ where $T^j_{2k+j,s}(\tau) = T^j_{2k+j+1}(\alpha_{2k+j+1} * 0)$ and $T^{1-j}_{n_{s+1},s+1} = Ext(T^{1-j}_{2k+j,s}, \alpha_{2k+j,s} * 01)$. All trees with smaller indices remain as at stage s. In this case, we satisfy the kth minimality requirement for A^j as there are no k–splittings on $T^j_{2k+j,s+1}$ above $\alpha_{j,s+1}$. We also record the fact in A^{1-j} that we switched from a k–splitting subtree by forcing $\alpha_{1-j,s+1}$ off the narrow subtree of $T^{1-j}_{2k+j,s+1}$.

If no $i < k$ is enumerated and $2k + j + 1 = n_s + 1$ we extend our sequences of trees by setting $n_{s+1} = n_s + 1$ and

$$T^j_{n_s+1} = Nar(T^j_{n_s})$$

$$T^{1-j}_{n_s+1} = Sp(T^{1-j}_{n,s}, k).$$

Clearly the number of times a tree $T^j_{i,s}$ is changed is finite and so we can argue as usual that the A_j are of minimal degrees. We claim that we can compute $\varnothing''$ and recover the sequence (of indices for) $T^j_i = \lim T^j_{i,s}$ from $A'_0 \oplus A'_1$. Suppose we have $T^0_0, \ldots, T^0_{2i}$ and $T^1_s, \ldots, T^1_{2i}$ and have computed $\varnothing'' \restriction i$ and a stage s_{2i} by which all of these have settled down. We ask A'_1 if A_1 is on $Nar(T^1_{2i})$. If so, $i \notin \varnothing''$, $T^0_{2i+1} = Sp(T^0_{2i})$, $T^1_{2i+1} = Nar(T^1_{2i})$ and all have settled down by $s_{2i+1} = s_{2i}$. If not, we wait for a stage s at which we forced A_1 off $Nar(T^1_{2i})$. At stage s, we set $T^1_{2i+1,s} = Nar(Ext(T^1_{2i}, \tau))$ for some τ (possibly $\varnothing$). We again ask A'_1 if A_1 is on $T^1_{2i+1,s}$. If so, T^j_{2i+1} has reached its limit for $j = 0, 1$ as has $\varnothing''(i)$. If not, we find $s' > s$ at which we force A_1 off $T^1_{2i+1,s}$. As we can do this at most twice, we then have $T^j_{2i+1,s'} = T^j_{2i+1}$ and $\varnothing''(i) = \varnothing''_{s'}(i)$. We can now find T^j_{2i+2} by asking of A'_0 if A_0 is on $Nar(T^0_{2i+1})$. If so, T^j_{2i+2} are already fixed and if not they become fixed when we force A_0 off $Nar(T^0_{2i+1})$. $\square$

References

Cooper, S. B., *Minimal degrees and the jump operator*, J. Symbolic Logic **38** (1973), 249–271.

Epstein, R. L., *Minimal degrees of unsolvability and the full approximation construction*, Memoirs AMS **162** (1975).

Friedberg, R. M., *A criterion for completeness of degrees of unsolvability*, J. Symbolic Logic **22** (1957), 159–160.

Jockusch, C. G. Jr. and Posner, D. B., *Double jumps of minimal degrees*, J. Symbolic Logic **43** (1978), 715–724.

Lerman, M., *Degrees of unsolvability*, Springer-Verlag, Berlin, 1983.

Posner, D. B., *A survey of non-r.e. degrees* $\leq \mathbf{0}'$, Recursion Theory: Its Generalizations and Applications (Proc. Logic Colloquium 1979, Leeds, August 1979) (F. R. Drake and S. S. Wainer, eds.), London Math. Soc. LNS **45**, Cambridge University Press, Cambridge, England, 1980.

Robinson, R.W., *Interpolation and embedding in the recursively enumerable degrees*, Ann. of Math (2) **93** (1971), 586–596.

Sacks, G. E., *A minimal degree below* $\mathbf{0}'$, Bull. AMS (N.S.) **67** (1961), 416–419.

______, *Recursive enumerability and the jump operator*, Trans. AMS **108** (1963), 223–239.

______, *On the degrees less than* $\mathbf{0}'$, Ann. Math. (2) **77** (1963), 211–231.

Sasso, L., *A minimal degree not realizing the least possible jump*, J. Symbolic Logic **34** (1974), 571–573.

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

______, *A theorem on minimal degrees*, J. Symbolic Logic **31** (1966), 539–544.

______, *Degrees of Unsolvability*, Math. Studies **2**, North-Holland, Amsterdam, 1971.

Simpson, S.G., *Minimal covers and hyperdegrees*, Trans. AMS **209** (1975), 45-64.

Soare, R. I., *Recursively Enumerable Sets*, Springer-Verlag, Berlin, 1987.

Spector, C., *On degrees of recursive unsolvability*, Ann. Math. (2) **64** (1956), 581-592.

Yates, C. E. M., *Initial segments of the degrees of unsolvability, Part II: Minimal degrees*, J. Symbolic Logic **35** (1970), 243–266.

______, *Prioric games and minimal degrees below* $\mathbf{0}'$, Fund. Math. **82** (1974), 217–237.

Department of Mathematics, Victoria University of Wellington, Wellington, New Zealand; Department of Mathematics, University of Wisconsin, Madison, WI 53706-1388, USA; Department of Mathematics, Cornell University, Ithaca, NY 14853-7901, USA

E-mail address: downey@math.vuw.ac.nz; lempp@math.wisc.edu; shore@math.cornell.edu