Recursively Enumerable $m$- and $tt$-Degrees II: The Distribution of Singular Degrees

R. G. Downey

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

1. Introduction

In [2], Degtev constructed what we call a singular $tt$-degree, namely an r.e. $tt$-degree containing a single r.e. $m$-degree. It is quite easy to construct an r.e. $tt$-degree containing infinitely many r.e. $m$-degrees [9], and in [4], the author constructed an r.e. $tt$-degree with no greatest r.e. $m$-degree.

In Part I of this paper [5], the author solved a question of Odifreddi [16, Problem 10] by showing that it is possible for an r.e. $tt$-degree to contain a finite number of r.e. $m$-degrees and yet not be singular. In particular, for any finite nonzero $n$, in [5] the author constructed an r.e. $tt$-degree containing exactly $2^n - 1$ r.e. $m$-degrees.

In the present paper we turn to more "global" aspects by analysing the distribution of singular $tt$-degrees within r.e. $T$-degrees. Our investigations were inspired by Odifreddi [17], who—in particular—asked if $\mathcal{O}_f$ contains a singular r.e. $tt$-degree.

Define an r.e. $T$-degree $a$ to be singular if $a$ contains a singular r.e. $tt$-degree.

In Sect. 2 we show that $\mathcal{O}_f$ is singular. In Sect. 3 we show that singular r.e. $T$-degrees are dense in the r.e. $T$-degrees. Finally, in Sect. 4 we show that not all r.e. $T$-degrees are singular. In fact we construct an r.e. $T$-degree $a$ such that if $b$ is an r.e. $tt$-degree contained in $a$, then $b$ has no greatest r.e. $m$-degree. In Sect. 5 we close with a few open questions.

Notation and terminology are standard. A good reference is Soare [21]. The following are exceptions. The upper case Greek letters $\Gamma$ and $\Lambda$ are reserved for $tt$-reductions with use functions $\lambda$ and $\gamma$ respectively. We remind the reader that the use function $\lambda$ of a $tt$-reduction $A$ is defined as the largest member of the truth table describing $A$. (Thus if $A$ is the reduction given by the recursive function $f$ then $x$-use of $A$ is the largest member of the $f(x)$th truth table.) Note that, without loss, we can take all use functions as monotone increasing both in argument and stage number (where defined). The upper case Greek letters $\Phi$ and $\Psi$ are reserved for $T$-functionals. We let $\{\delta_e\}_{e \in \omega}$ denote a standard enumeration of all partial recursive unary functions. All computations, etc., are bounded by $s$ at stage $s$. We
warn the reader that the argument of Sect. 3 relies on tree of strategies arguments along the lines of Lachlan [14] or Soare [20, 21]. It is helpful but not essential to be familiar with Sect. 2 of Part I of this paper. Finally we denote \( \{z : z \in A \& z \leq x\} \) by \( A[x] \).

The author thanks George Odifreddi for helpful discussions. He also thanks the referee for such an excellent and helpful report. [See also the remark after the proof of (4.1).]

2. A \( T \)-Complete Singular tt-Degree

In this section, we solve a question of Odifreddi [17] by showing that \( O' \) is singular. Interestingly, although the construction [2] (or [5]) of a singular degree is finite injury, to show that \( O' \) is singular requires an infinite injury argument. We build \( A = \bigcup_i A_i \) with \( A = \tau K \) and satisfying

\[
R_e : A_e(A) = V_e \quad \text{and} \quad \Gamma_e(V_e) = A \quad \text{implies} \quad V_e \equiv_a A.
\]

Here \( R_e \) is considered for all triples \( (A_{\omega}, \Gamma_e, V_{\omega_0}) \) consisting of an r.e. set \( V_e \) and two \( tt \)-reductions \( A_{\omega}, \Gamma_e \) with uses \( \lambda_e \) and \( \gamma_e \) respectively. Let \( f(\omega) = K \) be a recursive 1-1 enumeration of \( K \). At each stage \( s \) we let \( \{a_{i,s} : i \in \omega\} \) enumerate \( A_{i,s} \) in increasing order. We ask that at each stage \( s \), \( A_{s+1} = A_s \cup \{a_{j,s} : k \leq j \leq k+s \} \) for some \( k \leq f(s) \). (The reason for the "dump" becomes clear later.) The reader can check

\[(2.1) \text{Lemma.} \text{If } \lim s a_{i,s} = a_i \text{ exists then } A = \tau K.\]

Associated with the \( R_e \) is a restaint \( r(e, s) \). We shall assure that \( \lim \inf_s r(e, s) < \infty \) where \( R(e, s) = \max \{r(j, s) : j \leq e\} \). To meet the \( R_e \) the basic strategy is sketched below (see [5] for more details).

Let \( L(e, s) = \max \{x : \forall y < x(A_{e,s}(A_{i,s}) y = V_{e,s}(y))\} \) and

\[l(e, s) = \max \{x : \forall y < x(A_{e,s}(A_{i,s}) y = V_{e,s}(y)) & L(e, s) > \gamma_s(y))\}.\]

The reader should think of \( l(e, s) \) as the "\( A \)-controllable" length of agreement. Let \( A_i = A_i \cup \{a_{j,s} : i \leq j \leq i+s\} \). To satisfy a single \( R_e \) we monitor \( l(e, s) \). When \( l(e, s) > a_{i,s} \) we ask questions of the \( tt \)-reductions \( A_{e,s} \) and \( \Gamma_e \) concerning the effect of possible future configurations of \( A \). By the dump property, these can only be of the form \( A_j \) for \( j \leq i \) (with \( j \) not restrained by higher priority requirements).

Thus for example setting \( A_{i+1} = A_i \) must cause a change in \( V_{e,s} \) below \( \gamma_s(a_{const}) \). What we look for is an inconsistency. Thus, for example, an initial segment of \( A \) looking like \( A_j \) might predict \( y(i) \) enters \( V_e \) (to remain consistent) and yet \( A_i \) for some \( j < i \) might predict \( y(i) \) doesn't enter \( V_e \) but some \( y(j) \neq y(i) \) does. The key idea is that in this case we can force a disagreement by first setting \( A_{i+1} = A_j \) and then, when we see \( y(i) \) enter \( V_e \) setting \( A_{i+1} = A_i \) and preserving this initial segment of \( A \).

Roughly speaking, we can argue that if it is impossible to kill \( R_e \) as above then we can tie single numbers entering \( A \) to single numbers entering \( V_e \). Fuller intuitive remarks may be found in [5].

Coding \( K \) into \( A \) presents certain rather nasty problems since for all attacks, once we have set up a disagreement, still later \( K \)-coding can injure this allowing the computations to correct themselves. In particular the problems are sufficiently complex that we need \( R_e \) to be perhaps injured infinitely often and still be able to
sively Enumerable m- and tt-Degrees II

\[ \text{for } j > e. \text{ Roughly speaking, we do this by determining which requirement is satisfied by the point of } A_e \text{ at which we can attack requirements. This is why we say that } i \text{ requires attention (see below) rather than } R_e \text{ requires attention. We turn to the construction:} \]

**Definition.** We shall say that \( i \) **requires attention via \( e \)** at stage \( s+1 \) if \( e \) is least that \( R_e \) is not currently (declared) satisfied, \( i > \max \{ e, r(j, s) : j < e \} \) and one of the options below holds:

**Option.** For all sets \( \bar{V}_e \) with \( V_{e,s} \subseteq \bar{V}_e \) and \( \bar{V}_e \subseteq \{0, \ldots, s\} \) we have \( \Gamma_e(\bar{V}_e)[l(e,s)] \) \([l(e,s)] \text{ or } \bar{V}_e[m] = A_e(A_i)[m] \) where \( m = \max \{ \gamma_{e,s}(z) : z \leq l(e,s) \} \).

**ark.** (2.3) says that we can change \( A_{e+1} \) to \( A_i \) then \( V_{e,s} \) can't respond to preserve \( A_e \).

**Option.** We have that \( a_{i+1,s} < l(e,s) \) and we can win in a two step action based \( a_{i+1,s} \) then \( a_{i,s} = : \text{ that is, for all sets } \bar{V}_e \subseteq \{0, \ldots, s\} \) if

(i) \( J_{e,s}(\bar{V}_e) \subseteq A_{e} \) or \( \bar{V}_e[m] = A_e(A_i)[m] \).

(ii) \( J_{e,s}(\bar{V}_e) \subseteq A_{e} \) denotes what \( V_{e,s} \) will be if we set \( A_{e+1} = A_i^{i+1} \). That is, if we let \( my(e,s) = \{ \gamma_{e,s}(y) : y < l(e,s) \} \) then \( J_{e,s} = A_{e,s}(\gamma_{e,s}[m]) \) and where \( \max \{ \gamma_{e,s}(z) : z < l(e,s) \} \).

**ark.** (2.4) says that if \( I \) first set \( A \) to be \( A_i^{i+1} \) wait for recovery of the \( I_e; A_e \) putations then \( I \) can win by then setting \( A \) to be \( A_i \) as in (2.3).

**Definition.** We say that \( i \) **demands attention at stage \( s+1 \)** if there exists an \( e \) that

(i) \( f(s) \geq i+1 \),

(ii) \( i \) requires attention via \( e \), and

(iii) \( e \) is the least such.

the least \( e \) by which \( i \) requires attention we say \( i \) demands attention via \( e \).

**struction**

\( e < 0 \)

\( A_0 = \emptyset \). Initialize all \( R_e \) (meaning that \( R_e \) is unsatisfied and \( r(e,0) = 0 \)).

\( e > 0 \)

if any \( i \) demands attention. If not set \( A_{e+1} = \{ a_f(0,s), \ldots, a_f(0,s) \} \). If this ates any restraints cancel those restraints and initialize the requirement to ch the restraints pertained. [Of course a restraint \( r(e,s) \) is violated if \( j < r(e,s) \) \( a_{j,s} \) enters \( A_{e+1} - A_e \)]

If \( i \) demands attention adopt the appropriate case below:

**e 1.** (2.3) holds. Set \( A_{e+1} = A_i \). Set \( r(e,s+1) = i \). Declare \( R_e \) as (temporarily) satisfied.

**e 2.** (2.4) holds. Set \( A_{e+1} = A_i^{i+1} \). Set \( r(e,s+1) = i+1 \).

Either case initialize all those restraints whose restraints are violated.

1 of Construction.
Verification

(2.6) Lemma. $\lim_s a_{e,s} = a_e$ exists and $A = \tau K$.

Proof. Let $s_0$ be a stage such that for all $s > s_0$ for all $j < e$ we have

(i) $a_{j,s} = a_{j,s_0} = a_j$

(ii) $f(s) > e$.

Now $a_{e,s}$ can only further change due to $e$ demanding attention, or $e-1$ demanding attention [via (2.4)]. Let $R_k$ be the highest priority requirement with $e-1$ demanding attention via $k$ after stage $t$. When $R_k$ pertains to $a_{e-1}$ at some least stage $s_1$ we set $r(k,s_1) = e$, initialize all $R_k$ for $k < k$ and set $A_{s_1+1} = A_s e$. We claim that

(2.7) either $\forall t > s_1 (l(k,t) < a_{e-1})$, or

(2.8) if $t = \mu s > s_1 (l(k,s) > a_{e-1})$ then for all $\tilde{t} > t$ there exists $\tilde{k} = \tilde{k}(t) < k$, $r(\tilde{k}, \tilde{t}) = r(k,t) = e$.

Now when $e-1$ demanded attention at stage $s_1$, we reset $a_{e,s_1+1}$ to exceed the use of the $l(k,s_1) > a_{e-1}$ computations. By choice of $s_0$ and monotonicity of the $\{a_{e,s} = g \in \omega\}$, it follows that for all $t > s_1$, $l(k,t) > a_{e-1}$. Now when $e-1$ demanded attention at stage $s_1$, (2.4) pertains to $e-1$. By (2.4)(iii), $V_{k,t}$ must be an extension of $J_{k,t}$, and hence if ever $r(\tilde{k}, \tilde{t}) < e$ for all $\tilde{k} < k$ at some stage $t_1 > t$ it follows that (2.3) will hold for $e-1$ and $k$. Thus $e-1$ will demand attention via some $\tilde{k} < k$ and hence by $k$ (hypothesis). But now (2.3) pertains to $e-1$ and we would set $A_{t_1+1} = A_{t} e$ contradicting choice of $s_0$. Thus either (2.7) or (2.8) holds.

Let $t = s_1$ if (2.8) holds, let $q = s_1$ if (2.7) holds and let $q = s_0$ if $e-1$ does not demand attention after stage $s_0$. Then after stage $q$, $a_{e,s}$ can only further change if $e$ demands attention via some $k$ and (2.3) pertains. Now $A_0$ be the highest priority requirement such that $e$ demands attention via $d$ after stage $q$, say at stage $q_1$. It is really quite easy to see that $a_{e,q_1+1} = a_{e,s}$, for all $r > q_1$. Note that this follows because the restraint $r(d,q_1+1) = e$ is never violated after stage $q_1$ and so never cancelled.

Hence $\lim_s a_{e,s} = a_e$ exists in any case and $A = \tau K$ by (2.1). □

(2.9) Lemma ("Window Lemma"). $\lim_s R(e,s) = R(e)$ exists.

Proof. Suppose that for all $j < e$, $\lim_s R(j,s) = R(j)$ exists. If $N_e$ receives attention only finitely often then $R(e,s)$ is only reset finitely often and then it is obvious that $\lim_s R(e,s) = R(e)$ exists.

Thus, without loss, suppose that $N_e$ receives attention infinitely often. This means that $\lim_s R(e,s) = R(e)$ exists, and $R(e) = 0$. But we must establish this for $R(e)$. We claim that $\lim_s R(e,s) = \lim_s R(e-1,s)$. Let $s_1$ be a stage where $r(e,s_1) = R(e-1)$. Now at some least stage $s_2 > s_1$, we must initialize $r(e,s_2)$ from $r(e,s_1)$ to zero. If, at this stage, $R(e,s_2) > R(e-1)$, it can only be that for some $e_1 < e$ we have $r(e_1,s_1) > R(e-1)$. The crucial point is now that at any stage $s_1 > s_2$ whenever $r(e_1,s_3) = r(e_1,s_2)$ is initialized (as it must be since $\lim_s R(e_1,s) = R(e)$) $\leq R(e-1)$ exists. Thus $r(e,s_3)$ must also be the case that $r(e,s_3)$ is initialized too since it must respect $r(e_1,s_2)$ so long as the $r(e_1,s_2)$ restraint has been in force. Continuing this reasoning when $r(e_1,s_3)$ is initialized we either have $R(e,s_3)$
= R(e−1) or for some e2 < e1 we have r(e2, s2) > R(e−1), etc. This process is well founded and hence lim inf, R(e, s) = R(e−1) as required.

(2.10) Lemma. Suppose (2.3) pertains to i and e at stage s and we set r(e, s) = i. Suppose further that a_i,s = a_e. Then R_e is met at stage s (with l(e, s) → ∞).

Proof: r(e, s) is not violated and inspection of (2.3).

(2.11) Lemma. R_e is met.

Proof: By (2.10) we know that each time R_e is attacked such restraint must be violated by either coding or R_e for e < e. Let R(e) = lim inf,s R(e, s). We suppose that l(e, s) → ∞. We must show A ≡_m V_e.

Let s_0 be a stage such that for all s > s_0 for all i ≤ max{R(e), e}, a_i,s = a_i,s_0 = a_i. V_e ≡_m A: let z be given. Compute a stage s = s(z) > s_0 such that r(e, s) = 0 and R(e, s) = R(e) and l(e, s) > z. Since γ_i(z) ≥ z by standard convention, it must be that λ_i(z) is defined and L(e, s) > z. Now define i to be attainable if i > max{e, R(e)}. Without loss, choose z_i ∈ V_e, see if for any attainable i we have that A_i(A_i) = z ∈ V_e. (That is the A_i truth table with oracle A_i say that z ∈ V_e.

If i does not exist, then choose it maximally. There are two cases.

Case 1. For all attainable j < i, A_i(A_i) = z ∈ V_e. In this case z ∈ V_e if a_i,s ∈ A by the dump property.

Case 2. ∃ j < i (A_i(A_i) = z ∈ V_e). In case 2 we must have A_i,s = A_i,s for some k ≤ i (at worst, i−1 will demand attention). Hence by (2.6), case 2 can apply to at most finitely many s. Thus search for the least stage t where z ∈ V_e, or z ∈ V_e and either z ∈ V_e cannot be forced or case 1 pertains. More formally, there are several subcases.

Subcase (a). A_i(A_i) = z ∈ V_e and for all attainable k ≤ i, A_i(A_i) = z ∈ V_e. Then z ∈ V_e.

Subcase (b). A_i(A_i) = z ∈ V_e but there exists k < k with A_i(A_i) = z ∈ V_e. In this case it must be that k−1 is attainable k−1 > R(e) and so this attack fails. Thus, compute a stage s_1 > s such that r(j, s) is first violated. At this stage we have A_i,s [s] = A_i,s [s] for some n ≤ k−1. By abuse of notation we write A_i,s = A_i,s−1. Compute the least stage s_2 ≥ s_1 such that l(e, s_2) > z. The point is that if z ∈ V_e then if there exists a (greatest) attainable j < s_2 with A_i(A_i) = z ∈ V_e it must be that (now) for all attainable j < s_1 with j < s_2 then we would have received attention at stage s. Thus in this case z ∈ V_e which satisfies A_i,s ∈ A.

Subcase (c). A_i(A_i) = z ∈ V_e. By the reasoning above if z ∈ V_e then if there exists a greatest attainable a_i,s, with A_i(A_i) = z ∈ V_e it must be that for all j < s_1, A_i(A_i) = z ∈ V_e too. Hence z ∈ V_e if a_i,s ∈ A.

Thus in case 2 also, we can effectively compute an element y(z) for z such that z ∈ V_e if y(z) ∈ A. Hence V_e ≡_m A.

A ≡_m V_e: this is very similar. Again compute a stage s = s(z) with l(e, s) > z and r(e, s) = 0 and R(e, s) = R(e) and s > s_0. Without loss z > a R(e) and z ∈ A_e. Define i to
be attainable as in $V_e \leq_m A$. Fix $a_{t+1} = z$. First if $i$ is not attainable then $z \notin A$. If $i$ is attainable compute a stage $s > t$ with \( r(e, t) = 0 \) and \( R(e, t) = R(e) \) and \( l(e, s) > a_{t+1} \). Without loss $z = a_{t+1} \notin A$. Now there are two subcases.

**Subcase 1.** There is a (least) number $\hat{z} < my(e, t)$ such that

(i) $A_{\hat{z}} \cup (A_{\hat{z}+1}) = \hat{z} \notin V_e$ and

(ii) $A_{\hat{z}}, (A_{\hat{z}}) = \hat{z} \in V_e$ for all attainable $j \leq \hat{z}$.

In this case $z \in A$ iff $\hat{z} \in V_e$.

**Subcase 2.** Otherwise. In this case note that $i, i+1$ is a killing point for $N_e$, and either we set $A_{i+1} = A_i^k$ for some $k \leq i$ (in which case $z \in A$) or we set $A_{i+1} = A_e^l+1$ and attack some requirement $R_e$ for $l \leq e$. But now $z \in A$ since we know this attack fails. Hence $z \in A$ in this subcase.

This concludes the proof of $A \leq_m V_e$ and hence the whole lemma. \( \square \)

In Part I of this paper [5] the author constructed an r.e. tt-degree consisting of exactly $2^n - 1$ r.e. $m$-degrees (for any given $n > 1$). In the notation of [5], by encoding the block with the high e-state below the $f(s)$-th block if it is "s-dead", it is possible to extend the above to show

(2.12) **Theorem.** Let $n > 1$. Then \( O_T \) contains an r.e. tt-degree with exactly $2^n - 1$ r.e. $m$-degrees.

### 3. Density

In this section we prove

(3.1) **Theorem.** *Singular T*-degrees are dense in $R$.

**Proof.** The technique involved in the proof of this theorem consists of an amalgam of the basic strategy (for building a singular tt-degree) and a fairly standard coding and delayed permitting argument which has been seen in several other density type theorems (e.g., [7, 8, 10]). In consequence, we shall concentrate on the intuition of the construction and refer the reader elsewhere should he desire more formal details.

Let $E \preceq_T F$ be r.e. sets. By Sacks density theorem [19] it suffices to build an r.e. set $A$ with $E \leq_T A \leq_T F$ with the tt-degree of $A$ singular. The basic strategy remains the same; we keep examining $l(e, s)$ and if we see a one or a two step action to kill $E$ we act to do so. Thus $R_e$ "requires attention" if $e$ is least such that (2.3) or (2.4) pertains to some $i$ (least). (A subtle point here is that we have changed back to $e$ rather than $i$ requiring attention.)

In itself $F$-permitting causes no problems with this strategy. The first problem is that $E$-coding can injure infinitely often an apparently satisfied $R_e$ by enumerating elements below $r(e, s)$ (as in Sect. 2). The second problem is a "coherence" one — induced by our solution to the first — which we discuss later.

The solution to the first problem, not surprisingly, involves arranging matters to show that if $R_e$ fails to be met then $F \leq_T E$. That is we must attempt to build a reduction $\forall(E) = F$ predicated on $R_e$'s failure.

To be more specific we shall use the marker coding of Sect. 2. Namely at stage $s$, $A_{s+1} = A_s^k$ for some $k \leq f(s)$ where $f(o) = E$. Also to ensure $A \leq_T F$ we shall ask that
Recursively Enumerable m- and t-Degrees II

F permits elements to enter A. We refer to i as an e-attack point for \( R_a \) if \( R_a \) requires attention via i. Now although \( R_a \) may require attention via i, it may not be able to attack at i since additionally we need that (for the time being, anyhow) F permits on i at s. The first approximation is simply to attack any e-attack point in the manner of Sect. 2 whenever we get an F-permission. Implicitly here although we may believe \( R_a \) satisfied at i if we can attack \( R_a \) at \( \hat{i} < i \), we will do so since this attack is more likely to succeed. We see that for a single requirement this idea is enough.

(3.2) Lemma (One requirement). Suppose that \( l(e, s) \rightarrow \infty \). Then \( R_a \) is met.

Proof. Define a stage \( s \) to be E-correct for i if \( E_{s}[i] = E[i] \). The point about E-correct computations is that for a single \( R_a \) if we use an E-correct i as an attack point then such an attack will succeed.

We claim that there can only be finitely many E-correct attack points. Suppose otherwise. Then there must exist infinitely many E-correct i to which (2.4) pertains but only finitely many E-correct i to which (2.3) pertains. We show how to compute F from E. Let \( s_0 \) be a stage such that \( \forall s > s_0 \) if (2.3) pertains to i at stage s and not at stage \( s_0 \) then i is E-correct at s.

Let s be given. Compute E-recursively a stage \( s = s(e) > s_0 \) and an attack point \( i > z \) such that \( i > s_0 \) and

(i) s is E-correct for i, and

(ii) i is an attack point for \( R_a \) at stage s.

Notice that (2.4) must pertain to i since (2.3) did not pertain to i at stage \( s_0 \). Furthermore (2.4) can't act at i since then at the first stage t with \( l(e, t) > a_i \), after (2.4) acts we would have that (2.3) pertains to i (or some number \( \leq i \) not alive at stage \( s_0 \)). It thus follows that \( \forall s > s(e) \) (f(s) = i) since (2.4) can't act. Thus \( F_s[i] = F[i] \). Hence \( F \leq T E \).

A totally similar argument works to show that if (2.3) pertains infinitely often E-correctly then \( F \leq T E \) also. Therefore we have that there are only finitely many E-correct attack points.

Now the argument works with the same methods as Lemma (2.11). Let i be the largest E-correct attack point and go to a stage \( s_0 \) where \( a_i s_0 = a_i \) and \( l(e, s) > a_i \) [assuming \( l(e, s) \rightarrow \infty \)]. Define \( j \) to be attainable if \( j > i \). Now, for example, to compute A m-recursively from \( V_e \) find a stage s where \( l(e, s) > z \). Without loss \( z = a_i \), with \( j \) attainable. As in (2.11) compute a stage \( t > s \) such that either \( z \in A \) or \( l(e, t) > a_{j+1} \). Assuming the latter there are again the two subcases.

Subcase 1. There is a (least) number \( \hat{z} < my(e, t) \) with

(i) \( A_{s_0}(A_{j+1}) = \hat{z} \notin V_e \)

(ii) \( A_{s_0}(A_{j}) = \hat{z} \in V_e \) for all attainable \( i \leq j \).

In this case \( z \in A \) iff \( \hat{z} \in V_e \).

Subcase 2. Otherwise. In this case we note that \( j, j + 1 \) is an attack point for \( R_a \) but any attack must fail since \( j \) must be E-incorrect. Thus in this case, as in (2.11), \( z \notin A \).

The other direction \( (V_e \leq_m A) \) is totally similar and is left to the reader. □

Thus we have a way of meeting a single \( R_a \). The second problem we must face is whether or not the above strategies can be made to cohere for several \( R_a \). Specifically we must worry that an infinitely active \( R_a \) might fatally injure an \( R_a \) for
\( \epsilon > e \). It is clearly possible that \( R_\varepsilon \) can act \( E \)-incorrectly infinitely often. The problem is this \( \varepsilon \) might desire \( i \) as an attack point. For some \( j > i, R_\varepsilon \) might be being attacked at \( j \) and so \( j = r(e, s) > i \). Now at this stage \( F \) decides to permit on \( i \). Unfortunately, since \( r(e, s) > i \) we can't use \( i \) to attack \( R_\varepsilon \) since \( \varepsilon \) must respect \( r(e, s) \).

But now although \( r(e, s) \) is later destroyed (since it was \( E \)-incorrect to begin with) we have lost our chance on \( i \). Thus although we meet \( R_\varepsilon \) since there are only finitely many \( E \)-correct attack points for \( i \), perhaps the infinitely many \( E \)-incorrect attack points use their \( E \)-false restraints to restrain \( R_\varepsilon \) when it wishes to act.

The single key observation needed to overcome this difficulty is that \( E \) knows if an attack is \( E \)-correct or not and furthermore since \( E \leq_T F \), whatever \( E \) knows \( F \) knows. Thus what we do is to use delayed permitting. That is, if \( i \) is as above when we see \( F \) permit on \( i \) we declare \( i \) as \( F \)-permitted. Should \( r(e, s) \) drop (because of \( E \)-incorrectness) we then allow \( i \) to be attacked should \( i \) still be an attack point. The whole point is that \( A \) remains \( \leq_T F \) since \( F \) can decide if such an event will occur.

In general we can thus satisfy \( R_1 \) in \( R_0 \)'s environment by (as usual on a \( \Pi_2 \)-guessing tree of strategies) building two versions of \( R_1 \). The first guesses that \( R_0 \) acts only finitely often and so behaves like a finite injury argument. The second version of \( R_1 \) guesses that \( R_0 \) acts (\( E \)-incorrectly) infinitely often and knows that \( \lim \inf_i r(0, s) = 0 \). It thus waits to act (by delayed permitting) for \( r(0, s) \) to drop.

The reader should note that Lemma (3.2) still holds with delayed permitting in place of permitting (i.e., should this version of \( R_1 \) be the correct one for the construction "along the true path"). The necessary observation is that once we know which version of \( R_1 \) is correct we can prediccate our reduction procedures on this parameter.

It is important to note that although \( F \) can't decide which version of \( R_1 \) is correct (a \( Q'' \)-question) nevertheless \( A \leq_T F \). The whole point is that \( F \) can decide the fate of any particular attack point, because \( F \) can decide if the correct environment for us to attack a particular point will occur.

There are no problems with the coherence of \( n > 2 \) requirements and we leave any further formal details to the reader (see [7, 8, 10] for arguments of a similar type). \( \square \)

4. A Nonsingular \( T \)-Degree

The goal of this section is to construct a nonsingular r.e. \( T \)-degree. In fact we can prove:

(4.1) Theorem. There exists an r.e. \( T \)-degree \( a \) such that if \( b \) is an r.e. \( \mathfrak{t} \)-degree contained in \( a \) then \( b \) has no greatest r.e. \( m \)-degree.

Proof. It is probably easiest to approach this construction by first describing the way one can construct an r.e. \( \mathfrak{t} \)-degree without greatest r.e. \( m \)-degree [4]. To do this we construct an r.e. set \( Q \) together with auxiliary r.e. sets \( B_e = \bigcup B_{e,s} \) such that we satisfy

\[
R_{e,1}: R_{e}(Q) = W_e \text{ implies } B_e \leq_n Q \text{ and } \neg(B_e \leq_n W_e \text{ via } \delta_i).
\]
Here we are working over pairs \( (r_e, W_e) \) consisting of a partial recursive \( tt \)-reduction \( r_e \) with use \( y_e \) an r.e.
set \( W_e \) together with a partial recursive function \( \delta_e \). We ensure that \( B_e \subseteq_n Q \) via \( x \in B_e \iff x \) is a follower targeted for \( B_e \) by stage \( x \) and \( 2x \in Q \) and \( 2x+1 \notin Q \).

We remark that here and in the subsequent analysis of the \( \mathcal{R}_{e,t} \) the \( tt \)-reductions are based on the assumption that the hypothesis of the requirements hold.

(This will be modified in the full construction of a nonsingular \( T \)-degree.)

Followers of \( R_{e,t} \) can be active or passive. Activity indicates that we are squeezing \( R_{x,t} \) as follows. Let \( q(e,s) = \max \{ x : \forall y < x ( r_e(Q_{e,s}(y)) = W_{e,s}(y)) \} \). We wait till \( q(e,s) > x \) (our follower) and \( \delta_{e,s}(x) \) and \( q(e,s) > \delta_e(x) \).
Now if \( \delta_{e,s}(x) = W_{e,s} \) as \( x \notin B_{e,s} \)
we can win \( R_{e,s} \) in one stroke by simply restricting \( Q \) [2x + 1]. [For then \( x \notin B_e \) and \( \delta_e(x) \in W_e \) hence \( \neg (B_e \subseteq_n Q \text{ via } \delta_e) \).]

On the other hand if \( \delta_{e,s}(x) \notin W_{e,s} \) we must be more careful. First if \( \delta_e(x) \notin r_e(Q_{e,s} \cup \{2x\}) \) then we can win as above -- by enumerating \( x \) into \( B_e \) and \( 2x \) into \( Q_{e,s+1} \)
and restricting \( Q \) [\( \max \{2x+1, \gamma_e(y_e(x))\} \) \]. Now \( x \in B_e \) but \( \delta_e(x) \notin W_e \). The problem case is if \( \delta_e(x) \in r_e(Q_{e,s} \cup \{2x\}) \).
Now we win in a two step action. First we enumerate \( 2x \) into \( Q_{e,s+1} - Q_e \) restrain \( Q \) [\( \max \{2x+1, \gamma_e(y_e(x))\} \)]
and declare \( R_{e,s} \) as active. Now we wait until \( W_{e,s} \) responds with a change \[ i.e., \text{ till } q(e,s) > \delta_e(x) \text{ again}. \]
Note that now \( \delta_e(x) \in W_e \). This we can win by enumerating \( 2x+1 \) into \( Q \) and keeping \( x \) out of \( B_e \). Now \( x \notin B_e \) but \( \delta_e(x) \in W_e \) [after all \( W_e \) can't withdraw \( \delta_e(x) \) once it enters].

The crucial point we must recognize is that -- with the above strategy -- should \( R_{e,t} \) be injured by higher priority activity we must automatically enumerate \( x \) into \( B_e \) if \( x \) is active we have a pending commitment to attend \( B_e \subseteq_n Q_e \).
Nevertheless such injuries can occur at most a bounded number of times and a very gentle finite injury argument does the rest.

The reader should note that the above strategy forms sort of an "inner strategy" for the full requirements below.

We shall build \( A \) and \( B_e \) for \( e \in \omega \) satisfy

\[
\mathcal{R}_{e,t}: \text{If } \Psi_e(A) = Q_e \text{, } \Phi_e(Q_e) = A \text{ and } r_e(Q_e) = W_e \text{, then } B_e \subseteq_n Q_e \text{ and } \neg (B_e \subseteq_n W_e \text{ via } \delta_e).
\]

Here we work relative to triples \( \Phi_e, \Psi_e, r_e, W_e, Q_e \) consisting of two \( T \)-functionals \( \Phi_e, \Psi_e \), a \( tt \)-recursive \( r_e \) and a pair of r.e.
sets \( \{ W_e, Q_e \} \), together with a partial recursive function \( \delta_e \). Of course, again we build \( B_e \subseteq_n Q_e \) but this time predicated on all of the \( e \)-computations converging. We need the auxiliary functions:

\[
q(e,s) = \max \{ x : \forall y < x (r_e(Q_{e,s}(y)) = W_{e,s}(y)) \},
\]

\[
L(e,s) = \max \{ x : \forall y < x (\Phi_e(Q_{e,s}(y)) = A_e(y)) \},
\]

\[
\phi(x,e,s) = \max \{ u(\Phi_e(Q_{e,s}(y)) : y < x \} \text{ for } x < L(e,s), \]

\[
l(e,s) = \max \{ x : x < L(e,s) \text{ & } x < q(e,s) \}
\]

\[
\forall y (y \leq \phi(x,e,s) \rightarrow \Psi_e(A_y, y) = Q_{e,s}(y)) \&
\]

\[
\forall y (y < \max \{ y_e(x) : z \leq x \} \rightarrow \Psi_e(A_y, y) = Q_{e,s}(y)) \).
\]
The reader should think of \( l(e, s) \) as the \( A \)-controllable length of agreement. The following Diagram 1 may be useful for visualising \( l(e, s) \)

Diagram 1

We satisfy the \( \bar{R}_{e,i} \) as follows. First there will be a region \( \leq y \) for some \( y \) devoted to satisfy \( \bar{R}_{e,i} \) of higher priority. (This is only significant for the \( tt \)-reduction we build from \( Q_e \) to \( B_e \).) To satisfy \( \bar{R}_{e,i} \) we wait till \( l(e, s) > y \) and pick a prefollower \( x = s \) for \( \bar{R}_{e,i} \) targeted for \( A_x \). We cancel all lower priority followers etc. at this stage. At the least stage \( s_1 > s \) when we see \( l(e, s_1) > x \) we declare \( x \) as \( e \)-confirmed, cancel again and assign a follower \( \bar{x} = s_1 \) targeted for both \( B_e \) and \( A \). Finally when we see \( l(e, s_2) > \bar{x} \) we declare \( \bar{x} \) as \( e \)-confirmed and again cancel all lower priority followers. This gives the situation in Diagram 2 below.

Diagram 2

Note that the only numbers \( z \) left alive after this process with \( y < z < s_2 \) are \( x \) and \( \bar{x} \).

Now we can specify the \( tt \)-procedure \( B_e \leq_B Q_e \). Thus

\[
(4.2) \quad \bar{x} \in B_e \iff \begin{cases} Q_{e,s_2}[\phi(x, e, s_2)] = Q_e[\phi(x, e, s_2)] \text{ and } Q_{e,s_2}[\phi(\bar{x}, e, s_2)] \\ \neq Q_e[\phi(\bar{x}, e, s_2)]. \end{cases}
\]
It is hoped that the reader can see the resemblance of (4.2) to the \( \leq_{tt} \) of \( R_{e,i} \). Roughly speaking \( \{ z : \phi(x, e, s_2) \leq_{tt} \phi(\hat{x}, e, s_2) \} \) has the role of \( 2x \). Note that if we hold \( A_{t}[\hat{x}] \) fixed then \( Q_{e,i}[\phi(\hat{x}, e, s)] \) also remains fixed; and if \( A_{t}[\hat{x}] \) changes then so must \( Q_{e,i}[\phi(\hat{x}, e, s)] \). We use \( x \) and \( \hat{x} \) to induce appropriate changes.

For a single \( R_{e,i} \) our strategy is now clear. We wait till a stage \( t \) occurs where \( \delta_{r}(\hat{x}) \) and \( l(e, t) > \delta_{r}(\hat{x}) \). At this stage if \( \delta_{r}(\hat{x}) \in W_{e,i} \), we can meet \( \hat{R}_{e,i} \) by simply cancelling all lower priority followers and (thereby) restraining \( A_{t}[\hat{r}] \). A typical situation is given in Diagram 3 below.

![Diagram 3](image)

Note that all numbers between \( \hat{x} \) and \( t \) are cancelled. We now win since \( \hat{x} \notin B_{e} \), but \( \delta_{r}(\hat{x}) \in W_{e} \).

If \( \delta_{r}(\hat{x}) \notin W_{e} \), we act as follows. Enumerate \( \hat{x} \) into \( A_{t+1} - A_{t} \) cancel all lower priority followers and wait till \( l(e, \hat{r}) > \delta_{r}(\hat{x}) \) for some stage \( \hat{r} > t \). At such a stage \( \hat{r} \)
again cancel lower priority followers. We now examine the effect of this on \( \delta_{r}(\hat{x}) \).

Since \( I_{e} \) is a \( tt \)-reduction we can ask what has happened to \( Q_{e}[\gamma_{e}(\delta_{r}(\hat{x}))] \). In particular, we see if now we have \( \delta_{r}(\hat{x}) \in W_{e,i} \) or not. If \( \delta_{r}(\hat{x}) \notin W_{e,i} \) we can now win by enumerating \( \hat{x} \) into \( B_{e} \) and restraining \( A_{t}[\hat{r}] \). This would mean that \( \hat{x} \in B_{e} \) but \( \delta_{r}(\hat{x}) \notin W_{e} \). Note that we must promise not to add \( x \) into \( A_{t+1} \) unless we also add \( z \leq_{tt} y \) into \( A_{t} \). This is of course fine since the only reason we would do so is because some \( R_{e,i} \) of higher priority acts, necessarily on numbers \( \leq_{tt} y \). But such action is precisely the reason for the first clause – involving \( y \) of (4.2).

If \( \delta_{r}(\hat{x}) \in W_{e,i} \) we can similarly win by now enumerating \( x \) into \( A_{t+1} \). This causes a change in \( Q_{e} \) below \( \phi(x, e, s) \) and so allows us, with priority \( (e, t) \) to keep \( \hat{x} \) out of \( B_{e} \). In this case \( \hat{x} \notin B_{e} \) but \( \delta_{r}(\hat{x}) \in W_{e} \) (we restrain \( A[x - 1] \) here). Again we need to note that if some number gets into \( A_{t} \) below \( y \) then we must act to correct \( B_{e} \) in this case by enumerating \( \hat{x} \) into \( B_{e} \) in response to such a change.

This concludes the discussion of a single \( R_{e,i} \). The coherence of the strategies causes no problems since all of the outcomes are finitary and hence a single finite injury argument will suffice. We know that any attack on \( R_{e,i} \), will be completed unless it is cancelled by some \( R_{e,i} \) of higher priority. The point is that if we always attack the requirement of highest priority then attacks on \( R_{e,i} \) must involve prefollowers less than those of \( R_{e,i} \). The only thing we must remember to do is to
attend any past commitments. Thus, when $R_{e,i}$ sets up the described $tt$-reduction and perhaps $R_{e,1}$ cancels this we must still honour this commitment. For example suppose $R_{e,1}$ begins an attack when at a stage $s_i$ where $l(e, s_i)$ exceeded all higher priority (pre-) follows we assigned to it a prefolower $x(\hat{e}) = s_j$. This action may have cancelled some set up $y_i(e)$, $x(e)$, $\hat{x}(e)$ for $R_{e,i}$ but note in that case if $\hat{x}(e)$ is $e$-confirmed then $s_i$ exceeds that use of the $\hat{x}(e)$-computations. Nevertheless, although $x(e)$ and $\hat{x}(e)$ are cancelled we may need to later add $\hat{x}(e)$ into $B_e$ should the first option of (4.2) hold (due to $R_{f,j}$ of priority higher than $R_{e,i}$). Similarly if we get another $R_{e,i}$ set up $y_1(e), x_1(e)$ before $\hat{x}(e)$ is defined, we would have $x(f) \leq y_1(e)$ and a stage $t$ where $l(e, t) > y_1(e)$ before $x_1(e) = t$ was defined. Since the $R_{e,t}$ set up is complete at the stage $u$ where $\hat{x}(e)$ is defined, although we cancel $x_1(e)$ and $\hat{x}(e)$, $\hat{x}(e)$ exceeds $\hat{x}_1(e)$'s $e$-use and so $R_{e,1}$ cannot interfere with this set up unless $x(e)$ enters which will only cause us to enumerate $\hat{x}_1(e)$ into $B_e$.

In this way it is easily seen that the strategies cohere and will be met via a finite (bounded) injury argument. Remaining details are left to the reader.

**Remark.** At this stage, I wish to again express my gratitude to the referee who pointed out that the strategies cohere as above in a finite injury argument. The original argument forced them to cohere only with completed set-ups and used the $O^m$-machinery combined with that of [6] to achieve this.

5. Further Results, Open Questions

We do not know if (4.1) can hold for high r.e. degrees, but conjecture that it does not. Namely we conjecture that the singular degree construction combines with Martin-Cooper high permitting [e.g., 1] to ensure that each high r.e. $T$-degree contains a singular r.e. $tt$-degree. The exact classification of the nonsingular $tt$-degrees would seem a difficult question.

The reader should note that it is unclear if (4.1) blends with downward degree control. For example if we try to use permitting when we enumerate $\hat{x}$ into $A$, we don't know how $Q_e$ will respond, and hence at the stage $s$ where the computations recover have a decision as to whether or not to enumerate $x$ into $A$. However we must apparently make this decision at $s$ and don't seem to be able to wait for permission. Of course, with standard techniques one can show that all promptly simple degrees bound nonsingular r.e. $T$-degrees, but we do not know if all nonzero r.e. degrees bound nonsingular degrees.

Indeed it is not even clear if they can be cappable. This brings us to another very interesting theme in the study of strong reducibilities in $T$-degrees namely the extent to which the structure of strong reducibilities within a $T$-degree reflects its lattice theoretic properties in $R$. The best known example is the local distributivity imposed by contiguity as in [15], but there are other examples. I believe this is a lot of very interesting and fruitful material here and ask, for example, if nonsingularity has any lattice-theoretic ramifications in $R$.

References

17. Odifreddi, P.: Questions on tt-degrees. (Handwritten notes, 1985)

Received December 18, 1986/in revised form March 24, 1988