Arch. Math. Logic (1988) 27: 135–147

Archive for
Mathematical
Logic

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 $\boldsymbol{0}'_T$ contains a singular r.e. *tt*-degree.

Define an r.e. *T*-degree $\boldsymbol{a}$ to be singular if $\boldsymbol{a}$ contains a singular r.e. *tt*-degree.

In Sect. 2 we show that $\boldsymbol{0}'_T$ 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 $\boldsymbol{a}$ such that if $\boldsymbol{b}$ is an r.e. *tt*-degree contained in $\boldsymbol{a}$ then $\boldsymbol{b}$ 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 Λ and Γ are reserved for *tt*-reductions with use functions λ and γ respectively. We remind the reader that the use function λ of a *tt*-reduction Λ is defined as the largest member of the truth table describing Λ. (Thus if Λ is the reduction given by the recursive function f then x-use of Λ 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 Φ and Ψ 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 $\{z: z \in A \& z \leqq x\}$ 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 $A = \bigcup_s A_s$ with $A \equiv_T K$ and satisfying

$$R_e: \Lambda_e(A) = V_e \quad \text{and} \quad \Gamma_e(V_e) = A \quad \text{implies} \quad V_e \equiv_m A.$$

Here R_e is considered for all triples $(\Lambda_e, \Gamma_e, V_e)_{e \in \omega}$ consisting of an r.e. set V_e and two *tt*-reductions Λ_e, Γ_e with uses λ_e and γ_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 $\bar{A}_s$ in increasing order. We ask that at each stage s, $A_{s+1} = A_s \cup \{a_{j,s}: k \leqq j \leqq k+s\}$ for some $k \leqq f(s)$. (The reason for the "dump" becomes clear later.) The reader can check

(2.1) Lemma. *If* $\lim_s a_{i,s} = a_i$ *exists then* $A \equiv_T K$.

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

Let $L(e,s) = \max\{x: \forall y < x(\Lambda_{e,s}(A_s; y) = V_{e,s}(y))\}$ and

$$l(e,s) = \max\{x: \forall y < x(\Gamma_{e,s}(V_{e,s}; y) = A_s(y) \,\&\, L(e,s) > \gamma_e(y))\}.$$

The reader should think of $l(e,s)$ as the "A-controllable" length of agreement. Let $A_s^i = A_s \cup \{a_{j,s}: i \leqq j \leqq i+s\}$. 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 Λ_e and Γ_e concerning the effect of possible future configurations of A. By the dump property, these can only be of the form A_s^j for $j \leqq i$ (with j not restrained by higher priority requirements).

Thus for example setting $A_{s+1} = A_s^i$ must cause a change in $V_{e,s}$ below $\gamma_e(a_{i,s})$. What we look for is an inconsistency. Thus, for example, an initial segment of A looking like A_s^i might predict $y(i)$ enters V_e (to remain consistent) and yet A_s^j 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 $A_{s+1} = A_s^i$ and then, when we see $y(i)$ enter V_e, setting $A_{t+1} = A_t^j$ 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 'y R_j for $j>e$. Roughly speaking, we do this by determining which requirement tack by the point of $\bar{A}_s$ at which we can attack requirements. This is why we say that i requires attention (see below) rather than R_e 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 options below holds:

Option. For all sets $\hat{V}_e$ with $V_{e,s}\subset\hat{V}_e$ and $\hat{V}_e\subset\{0,\ldots,s\}$ we have $\Gamma_e(\hat{V}_e)[l(e,s)]$ $[l(e,s)]$ or $\hat{V}_e[m]\neq A_e(A_s^i)[m]$ where $m=\max\{\gamma_{e,s}(z): z\leqq l(e,s)\}$.

ark. (2.3) says that we can change A_{s+1} to A_s^i then $V_{e,s}$ can't respond to preserve ements.

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 $a_{i,s}$": that is, for all sets $\hat{V}_e\subset\{0,\ldots,s\}$ if
(i) $J_{e,s}\subset\hat{V}_e$ then
(ii) $\Gamma_{e,s}(\hat{V}_e)\neq A_s^i$, or $\hat{V}_e[m]\neq A_e(A_s^i)[m]$.
iii) $J_{e,s}$ denotes what V_e will be if we set $A_{s+1}=A_s^{i+1}$. That is, if we let $m\gamma(e,s)$ ax$\{\gamma_e(y): y<l(e,s)\}$ then $J_{e,s}=A_{e,s}(A_s^{i+1})[m\gamma(e,s)]$ and where max$\{\gamma_{e,s}(z): z<l(e,s)\}$.

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

Definition. We say that *i demands attention at stage $s+1$* if there exists an e that
(i) $f(s)\geqq i+1$,
(ii) i requires attention via e, and
iii) i is the least such.
the least e by which i requires attention we say i demands attention *via e.*

struction

ge 0

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

ge $s+1$

if any i demands attention. If not set $A_{s+1}=A_s\cup\{a_{f(s),s},\ldots,a_{f(s)+s,s}\}$. 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 $A_{s+1}-A_s$.]
If i demands attention adopt the appropriate case below:

e 1. (2.3) holds. Set $A_{s+1}=A_s^i$. Set $r(e,s+1)=i$. Declare R_e as (temporarily) sfied.

e 2. (2.4) holds. Set $A_{s+1}=A_s^{i+1}$. Set $r(e,s+1)=i+1$.

either case initialize all those requirements whose restraints are violated.

l of Construction.

Verification

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

Proof. Let s_0 be a stage such that for all $s > s_0$ for all $j < e$ we have
(i) $a_{j,s} = a_{j,s_0} = a_j$ and
(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 $R_{\hat{k}}$ for $\hat{k} > k$ and set $A_{s_1+1} = A^e_{s_1}$. We claim that

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

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

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 $\{a_{g,s} = g \in \omega\}$, it follows that for all $\hat{t} > t$, $l(k, \hat{t}) > a_{e-1}$. Now when $e-1$ demanded attention at stage s_1, (2.4) pertained to $e-1$. By (2.4) (iii), $V_{k,t}$ must be an extension of J_{k,s_1} and hence if ever $r(\hat{k}, t_1) < e$ for all $\hat{k} < k$ 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 $\hat{k} \leqq k$ and hence by k (hypothesis). But now (2.3) pertains to $e-1$ and we would set $A_{t_1+1} = A^{e-1}_{t_1}$ contradicting choice of s_0. Thus either (2.7) or (2.8) holds.

Let $q = t$ 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 $\hat{k}$ and (2.3) pertains. Let R_d 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 $a_{e,q_1+1} = a_{e,r}$ 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 $A \equiv_T K$ by (2.1). □

(2.9) Lemma *("Window Lemma").* Lim $\inf_s R(e,s) = R(e)$ *exists.*

Proof. Suppose that for all $j < e$, lim $\inf_s R(j,s) = R(j)$ 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 R_e receives attention infinitely often. This means that lim $\inf_s r(e,s) = r(e)$ exists, and $r(e) = 0$. But we must establish this for $R(e)$. We claim that lim $\inf_s R(e,s) =$ lim $\inf_s R(e-1, s)$. 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, $R(e, s_2) \neq R(e-1)$, it can only be that for some $e_1 < e$ we have $r(e_1, s_2) > R(e-1)$. The crucial point is now that at any stage $s_3 > s_2$ whenever $r(e_1, s_3) = r(e_1, s_2)$ is initialized (as it must be since lim $\inf_s R(e_1, s) = R(e_1)$ $\leqq R(e-1)$ exists, by hypothesis) *it 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 $e_2<e_1$ we have $r(e_2,s_3)>R(e-1)$, etc. This process is well founded and hence $\lim\inf_s R(e,s)=R(e-1)$ 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* $a_{i,s}=a_i$. *Then* R_e *is met at stage s (with* $l(e,s)\to\infty$*).*

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 $R_{\hat{e}}$ for $\hat{e}<e$. Let $R(e)=\lim\inf_s R(e,s)$. We suppose that $l(e,s)\to\infty$. We must show $A\equiv_m V_e$.

Let s_0 be a stage such that for all $s>s_0$ for all $i\leqq\max\{R(e),e\}$; $a_{i,s}=a_{i,s_0}=a_i$. $V_e\leqq_m A$: 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 $\gamma_e(z)>z$ by standard convention, it must be that $\lambda_e(z)$ is defined and $L(e,s)>z$. Now define i to be *attainable* if $i>\max\{e,R(e)\}$. Without loss, choose $z\notin V_{e,s}$; see if for any attainable i we have that $\Lambda_e(A_s^i)\models z\in V_e$. (That is the Λ_e truth table with oracle A_s^i says that $z\in V_e$.)

If i exists, without loss, choose it maximally. There are two cases.

Case 1. For all attainable $j<i$, $\Lambda_e(A_s^j)\models z\in V_e$. In this case $z\in V_e$ iff $a_{i,s}\in A$ by the dump property.

Case 2. $\exists j<i(\Lambda_e(A_s^j)\models z\notin V_e)$.

In case 2 we must have $A_{s+1}=A_s^k$ for some $k\leqq i$ (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 $z\in V_{e,t}$ or $z\notin V_{e,t}$ and either $z\in V_e$ cannot be forced or case 1 pertains. More formally, there are several subcases.

Subcase (a). $\Lambda_e(A_s^k)\models z\in V_e$ and for all attainable $\hat{k}\leqq k$, $\Lambda_e(A_s^{\hat{k}})\models z\in V_e$. Then $z\in V_e$.

Subcase (b). $\Lambda_e(A_s^k)\models z\in V_e$ but there exists $\hat{k}<k$ with $\Lambda_e(A_s^{\hat{k}})\models z\notin V_e$. In this case it must be that $\hat{k}=k-1$ since if $\hat{k}<k-1$, $\hat{k}$ would demand attention (and so we'd set $A_{s+1}=A_s^{\hat{k}+1}$). This, in turn means that R_j for some $j\leqq e$ receives attention at $k-1$. The crucial point is that since $k-1$ 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 $A_{s_1}[s]=A_s^n[s]$ for some $n\leqq k-1$. By abuse of notation we write $A_{s_1}=A_{s_1-1}^n$. Compute the least stage $s_2\geqq s_1$ such that $l(e,s_2)>z$. The point is that if $z\notin V_{e,s_2}$ then if there exists a (greatest) attainable $\hat{\imath}<s_2$ with $\Lambda_e(A_{s_2}^{\hat{\imath}})\models z\in V_e$ it must be that (now) for all attainable $j<\hat{\imath}$, $\Lambda_e(A_s^j)\models z\in V_e$. Why is this? The point is that any such i must be $<n$ (by the dump property) and if there is an attainable $j<\hat{\imath}$ with $\Lambda_e(A_{s_2}^j)\models z\notin V_e$, then j would have received attention at stage s. Thus in this case $z\in V_e$ iff $a_{i,s_2}\in A$.

Subcase (c). $\Lambda_e(A_s^k)\models z\in V_e$. By the reasoning above if $z\notin V_e$ then if there exists a greatest attainable $a_{\hat{\imath},s}$ with $\Lambda_e(A_s^{\hat{\imath}})\models z\in V_e$ it must be that for all $j<\hat{\imath}$, $\Lambda_e(A_s^j)\models z\in V_e$ too. Hence $z\in V_e$ iff $a_{\hat{\imath},s}\in A$.

Thus in case 2 also, we can effectively compute an element $y(z)$ for z such that $z\in V_e$ iff $y(z)\in A$. Hence $V_e\leqq_m A$.

$A\leqq_m V_e$: 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 $z\notin A_s$. Define i to be attainable as in $V_e \leqq_m A$. Fix $a_{i,s} = z$. First if i is not attainable then $z \notin A$. If i is attainable compute a stage $t > s$ with $r(e,t) = 0$ and $R(e,t) = R(e)$ and $l(e,s) > a_{i+1,t}$. Without loss $z = a_{i,t} \notin A_t$. Now there are two subcases.

Subcase 1. There is a (least) number $\hat{z} < m\gamma(e,t)$ such that
(i) $A_{e,t}(A_t^{i+1}) \models \hat{z} \notin V_e$, and
(ii) $A_{e,t}(A_t^j) \models \hat{z} \in V_e$ for all attainable $j \leqq i$.
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 $A_{s+1} = A_s^k$ for some $k \leqq i$ (in which case $z \in A$) or we set $A_{s+1} = A_s^{i+1}$ and attack some requirement $R_{\hat{e}}$ for $\hat{e} \leqq e$. But now $z \in A$ since we know this attack fails. Hence $z \in A$ in this subcase.

This concludes the proof of $A \leqq_m V_e$ and hence the whole lemma. □

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 "σ-dead", it is possible to extend the above to show

(2.12) Theorem. *Let $n > 1$. Then $\mathbf{O}_T$ 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* $\mathbf{R}$.

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 $E <_T F$ be r.e. sets. By Sacks density theorem [19] it suffices to build an r.e. set A with $E \leqq_T A \leqq_T F$ 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 R_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 $F \leqq_T E$. That is we must attempt to build a reduction $\Psi(E) = F$ 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 $k \leqq f(s)$ where $f(\omega) = E$. Also to ensure $A \leqq_T F$ we shall ask that

F permits elements to enter A. We refer to i as an *e-attack point* for R_e if R_e requires attention via i. Now although R_e 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_e satisfied at i if we can attack R_e 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 $l(e,s)\to\infty$. Then R_e 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_e – 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-incorrect at s].

Let z be given. Compute E-recursively a stage $s=s(z)>s_0$ 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_e 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 $l(e,t)>a_{i,s}$ after (2.4) acts we would have that (2.3) pertains to i (or some number $\leqq i$ not alive at stage s_0). It thus follows that $\forall s>s(z)$ $(f(s)>i)$ since (2.4) can't act. Thus $F_s[i]=F[i]$. Hence $F\leqq_T E$.

A totally similar argument works to show that if (2.3) pertains infinitely often E-correctly then $F\leqq_T E$ 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 $l(e,s)\to\infty$]. 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 $z=a_{j,s}$ with j attainable. As in (2.11) compute a stage $t>s$ such that either $z\in A$ or $l(e,t)>a_{j+1,t}$. Assuming the latter there are again the two subcases.

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

(i) $A_{e,t}(A_t^{j+1})\models\hat{z}\notin V_e$

(ii) $A_{e,t}(A_t^{i})\models\hat{z}\in V_e$ for all attainable $i\leqq j$.

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_e, but any attack must fail since j must be E-incorrect. Thus in this case, as in (2.11), $z\in A$.

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

Thus we have a way of meeting a single R_e. The second problem we must face is whether or not the above strategies can be made to cohere for several $R_{\hat{e}}$. Specifically we must worry that an infinitely active R_e might fatally injure an $R_{\hat{e}}$ for

$\hat{e}>e$. It is clearly possible that R_e can act E-incorrectly infinitely often. The problem is this $\hat{e}$ might desire i as an attack point. For some $j>i$, R_e 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 $R_{\hat{e}}$ since $\hat{e}$ 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_e 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 $R_{\hat{e}}$ 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 $E \leqq_T F$, 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 $\leqq_T F$ 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 Π_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 $\liminf_s r(0,s)=0$. 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 predicate 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 $A \leqq_T F$. 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). □

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. tt-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. tt-degree without greatest r.e. m-degree [4]. To do this we construct an r.e. set Q together with auxiliary r.e. sets $B_e = \bigcup_s B_{e,s}$ such that we satisfy

$$R_{e,i}: \Gamma_e(Q)=W_e \text{ implies } B_e \leqq_{tt} Q \text{ and } \neg(B_e \leqq_m W_e \text{ via } \delta_i).$$

Here we are working over pairs (Γ_e, W_e) consisting of a partial recursive tt-reduction Γ_e with use γ_e, an r.e. set W_e together with a partial recursive function δ_i. We ensure that $B_e \leqq_{tt} Q$ 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 $\hat{R}_{e,i}$ – 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 $R_{e,i}$ can be *active* or *passive*. Activity indicates that we are squeezing $R_{e,i}$ as follows. Let $q(e,s) = \max\{x : \forall y < x(\Gamma_e(Q_s; y) = W_{e,s}(y))\}$. We wait till $q(e,s) > x$ (our follower) and $\delta_{i,s}(x)\downarrow$ and $q(e,s) > \delta_i(x)$. Now if $\delta_{i,s}(x) \in W_{e,s}$ as $x \notin B_{e,s}$ we can win $R_{e,i}$ in one stroke by simply restraining $Q[2x+1]$. [For then $x \notin B_e$ and $\delta_i(x) \in W_e$ hence $\neg(B_e \leqq_m Q \text{ via } \delta_i)$.]

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

The crucial point we must recognize is that – with the above strategy – should $R_{e,i}$ 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* $B_e \leqq_{tt} Q$. 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

$$\hat{R}_{e,i}: \text{If } \Psi_e(A) = Q_e, \Phi_e(Q_e) = A \text{ and } \Gamma_e(Q_e) = W_e$$
$$\text{then } B_e \leqq_{tt} Q_e \text{ and } \neg(B_e \leqq_m W_e \text{ via } \delta_i).$$

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

$$q(e,s) = \max\{x : \forall y < x(\Gamma_{e,s}(Q_{e,s}; y) = W_{e,s}(y))\},$$
$$L(e,s) = \max\{x : \forall y < x(\Phi_{e,s}(Q_{e,s}; y) = A_s(y))\},$$
$$\phi(x,e,s) = \max\{u(\Phi_{e,s}(Q_{e,s}; y)) : y < x\} \quad \text{for} \quad x < L(e,s),$$
$$l(e,s) = \max\{x : x < L(e,s) \;\&\; x < q(e,s)$$
$$\forall y(y \leqq \phi(x,e,s) \to \Psi_{e,s}(A_s; y) = Q_{e,s}(y)) \;\&$$
$$\forall y(y < \max\{\gamma_e(z) : z \leqq x\} \to \Psi_{e,s}(A_s; y) = Q_{e,s}(y))\}.$$

The reader should think of $l(e,s)$ as the A-controllable length of agreement. The following Diagram 1 may be useful for visualising $l(e,s)$

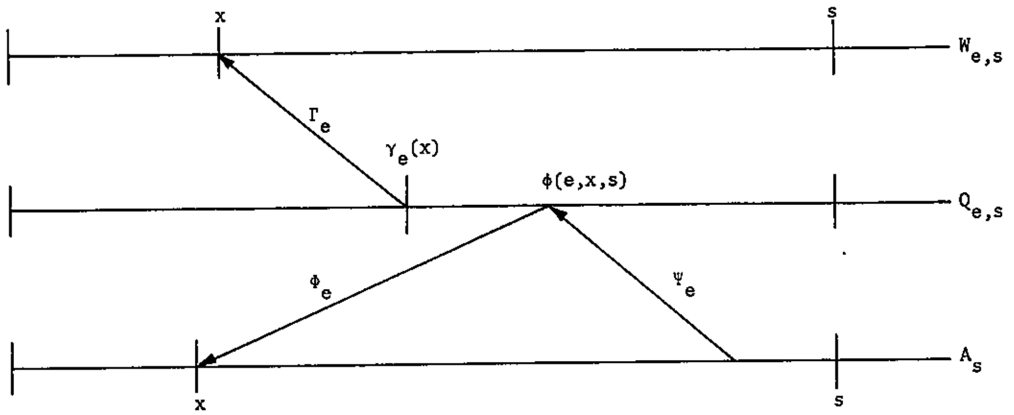

Diagram 1

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

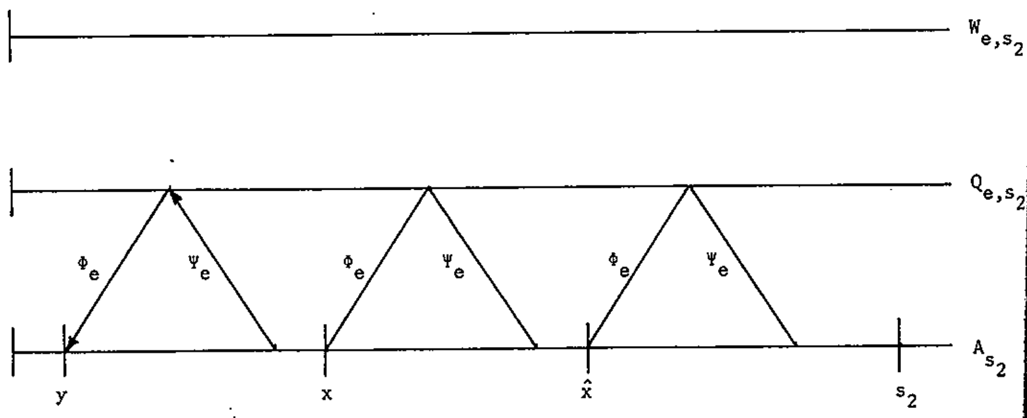

Diagram 2

Note that the only numbers z left alive after this process with $y<z<s_2$ are x and $\hat{x}$. Now we can specify the tt-procedure $B_e \leq_{tt} Q_e$. Thus

$$\text{(4.2)}\quad \hat{x}\in B_e \text{ iff } \begin{cases} Q_{e,s_2}[\phi(y,e,s_2)] \neq Q_e[\phi(y,e,s_2)] \text{ or} \\ Q_{e,s_2}[\phi(x,e,s_2)] = Q_e[\phi(x,e,s_2)] \text{ and } Q_{e,s_2}[\phi(\hat{x},e,s_2)] \\ \qquad \neq Q_e[\phi(\hat{x},e,s_2)]. \end{cases}$$

It is hoped that the reader can see the resemblance of (4.2) to the $\leq_{tt}$ of $R_{e,i}$. Roughly speaking $\{z:\phi(x,e,s_2)<z\leq\phi(\hat{x},e,s_2)\}$ has the rôle of $2x$. Note that if we hold $A_s[\hat{x}]$ fixed then $Q_{e,s}[\phi(\hat{x},e,s)]$ also remains fixed; and if $A_s[\hat{x}]$ changes then so must $Q_{e,s}[\phi(\hat{x},e,s)]$. We use x and $\hat{x}$ to induce appropriate changes.

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

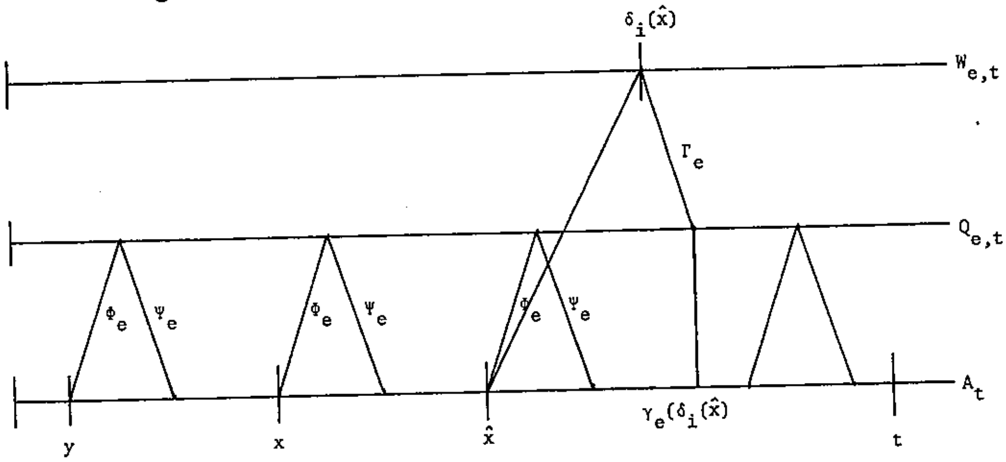

Diagram 3

Note that all numbers between $\hat{x}$ and t are cancelled. We now win since $\hat{x}\notin B_e$ but $\delta_i(\hat{x})\in W_e$.

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

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

If $\delta_i(\hat{x})\in W_{e,\hat{t}}$ we can similarly win by now enumerating x into $A_{\hat{t}+1}$. This causes a change in Q_e below $\phi(x,e,s)$ and so allows us, with priority $\langle e,i\rangle$ to keep $\hat{x}$ out of B_e. In this case $\hat{x}\notin B_e$ but $\delta_i(\hat{x})\in W_e$ (we restrain $A[x-1]$ here). Again we need to note that if some number gets into A 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 $\hat{R}_{e,i}$. 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 $R_{\hat{e},\hat{i}}$ of higher priority. The point is that if we always attack the requirement of highest priority then attacks on $\hat{R}_{\hat{e},\hat{i}}$ 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 $R_{\hat{e},\hat{i}}$ cancels this we must still honour this commitment. For example suppose $R_{\hat{e},\hat{i}}$ begins an attack when at a stage s_1 where $l(\hat{e}, s_1)$ exceeded all higher priority (pre-) follows we assigned to it a prefollower $x(\hat{e})=s_1$. This action may have cancelled some set up $y(e)$, $x(e)$, $\hat{x}(e)$ for $R_{e,i}$ but note in that case if $\hat{x}(e)$ is e-confirmed then s_1 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 $R_{\hat{e},\hat{i}}$). Similarly if we get another $R_{e,i}$ set-up $y_1(e)$, $x_1(e)$, $\hat{x}_1(e)$ before $\hat{x}(\hat{e})$ is defined, we would have $x(f) \leqq y_1(e)$ and a stage t where $l(e,t) > y_1(e)$ before $x_1(e)=t$ was defined. Since the $R_{e,i}$ set up is complete at the stage u where $\hat{x}(\hat{e})$ is defined, although we cancel $x_1(e)$ and $\hat{x}_1(e)$, $\hat{x}(\hat{e})$ exceeds $\hat{x}_1(e)$'s e-use and so $R_{\hat{e},\hat{i}}$ cannot interfere with this set up unless $x(\hat{e})$ 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'''-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 non-zero 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

1. Cooper, S.B.: Minimal Pairs and high recursively enumerable degrees. J. Symb. Logic 39, 655–660 (1974)
2. Degtev, A.N.: tt- and m-degrees. Algebra Logika **12**, 143–161 (Russian), 78–89 (English) (1973)
3. Degtev, A.N.: Three theorems on tt-degrees. Sib. Math. J. **17**, 1014–1022 (1976)

4. Downey, R.G.: Two theorems of truth table degrees. Proc. Am. Math. Soc. (in press)
5. Downey, R.G.: Recursively enumerable m- and tt-degrees I: the quantity of m-degrees. J. Symb. Logic (in press)
6. Downey, R.G., Jockusch, C.G.: T-degrees, jump classes and strong reducibilities. Trans. Am. Math. Soc. **301**, 103–136 (1987)
7. Downey, R.G., Slaman, T.: Completely mitotic r.e. degrees. Ann. Pure Appl. Logic (in press)
8. Fejer, P.: The density of the nonbranching degrees. Ann. Math. Logic **24**, 113–130 (1983)
9. Fischer, P.C.: A note on bounded truth table reducibility. Proc. Am. Math. Soc. **14**, 875–877 (1963)
10. Ingrassia, M.: P-genericity for recursively enumerable sets. Ph. D. Dissertation, University of Illinois, Urbana, 1981
11. Jockusch, C.G.: Relationships between reducibilities. Trans. Am. Math. Soc. **142**, 229–237 (1969)
12. Kallibekov, S.: On degrees of recursively enumerable sets. Sib. Math. J. **14**, 200–203 (1973)
13. Kobzev, G.N.: Relationships between recursively enumerable tt- and w-degrees. Bull. Akad. Sci. Georgian SSR **84**, 585–587 (1976)
14. Lachlan, A.H.: A recursively enumerable degree that will not split over all lesser ones. Ann. Math. Logic **9**, 307–365 (1975)
15. Ladner, R., Sasso, L.: The weak truth table degrees of recursively enumerable sets. Ann. Math. Logic **8**, 429–448 (1975)
16. Odifreddi, P.: Strong reducibilities. Bull. Am. Math. Soc., New Ser. **4**, 37–86 (1981)
17. Odifreddi, P.: Questions on tt-degrees. (Handwritten notes, 1985)
18. Odifreddi, P.: Classical recursion theory. Amsterdam: North-Holland (in press)
19. Sacks, G.E.: The recursively enumerable degrees are dense. Ann. Math., II. Ser. **80**, 300–312 (1964)
20. Soare, R.I.: Tree arguments in recursion theory and the $0'''$-priority method. In: Nerode, A., Shore, R. (eds.) recursion theory. Am. Math. Soc. Symposia **42**, 53–106 (1985)
21. Soare, R.I.: Recursively enumerable sets and degrees. (Omega Ser.) Berlin Heidelberg New York Tokyo: Springer 1987

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