Abstract

We show the decidability of the existential theory of the recursively enumerable degrees in the language of Turing reducibility, Turing reducibility of the Turing jumps, and least and greatest element.

The Existential Theory of the Poset of R.E. Degrees with a Predicate for Single Jump Reducibility

Steffen Lempp*
Department of Mathematics
University of Wisconsin
Madison, WI 53706

Manuel Lerman†
Department of Mathematics
University of Connecticut
Storrs, CT 06269

October 10, 1997

*Research partially supported by NSF Grants DMS-8701891, DMS-8901529, and INT-8722296 as well as by the Mathematical Sciences Research Institute.

†Research partially supported by NSF Grants DMS-8521843 and DMS-8900249 as well as by the Mathematical Sciences Research Institute.

1 Introduction

An important topic in classical recursion theory is the decidability of (fragments of) first-order theories of recursion-theoretic degree structures. The problem has been completely solved for the poset of the Turing degrees $\mathcal{D} = \langle \mathbf{D}, \leq \rangle$: Lachlan [9] showed the undecidability of the theory; Simpson [19] characterized the theory as being recursively isomorphic to second-order arithmetic; and Shore [16] and Lerman [13] showed that the $\forall\exists$-theory is decidable whereas Schmerl (cf. [13]) showed that the $\exists\forall\exists$-theory is undecidable. Similar sharp results have been obtained for the poset of the Δ_2-degrees $\mathcal{D}(\leq \mathbf{0}') = \langle \mathbf{D}(\leq \mathbf{0}'), \leq \rangle$ (by Epstein [4] and Lerman [13]; Shore [17]; Lerman and Shore [14]; and Schmerl (cf. [13])); as well as for some other structures.

For the recursively enumerable (r.e.) degrees, gaps remain. The theory of the poset of r.e. degrees $\mathcal{R} = \langle \mathbf{R}, \leq \rangle$ was shown to be undecidable by Harrington and Shelah [6] and to be of degree $\mathbf{0}^{(\omega)}$ by Harrington and Slaman [7]. While these results actually show the four-quantifier theory to be undecidable (personal communication), only the existential theory is known to be decidable by an easy modification of the Friedberg-Mučnik Theorem [5, 15]. The major obstacle to closing the gap is the characterization of all finite lattices embeddable into the r.e. degrees (cf. Ambos-Spies and Lerman [1, 2]).

Apart from Turing reducibility, the Turing jump operator is the most important symbol in any language of degree structures. The jump is definable from the partial order by Cooper [3], however not by an $\forall\exists$-definition by Lerman and Shore [14]. Thus the study of the Turing degrees with both partial order and jump is more complex. By an easy observation of Jockusch and Soare (cf. [13]), the theory of the Turing degrees with only the jump operator $\langle \mathbf{D}, ' \rangle$ is decidable; and by a recent forcing argument of Hinman and Slaman [8] outside the hyperarithmetic degrees, the existential theory of the Turing degrees with both partial order and jump $\mathcal{D}' = \langle \mathbf{D}, \leq, ' \rangle$ is decidable.

Our approach to including the jump operator in the language has been to use priority arguments. A natural first candidate is to show the decidability of the existential theory of the r.e. degrees in the language of partial order, least and greatest element, and predicates for nth jump reducibility ($\mathbf{a}^{(n)} \leq \mathbf{b}^{(n)}$). (Note that $\mathbf{a}^{(n)}$ and $\mathbf{b}^{(m)}$ are always comparable for r.e. degrees $\mathbf{a}$ and $\mathbf{b}$ if $n \neq m$.) The argument for this requires a new framework for $\mathbf{0}^{(n)}$-priority arguments for arbitrarily large n (cf. [10]), and we hope to present it in a future paper [11]. Our subsequent goal is to use the n-REA degrees to show the decidability of the existential theory of the Turing degrees with partial order, jump, and least element.

Here, we present a partial result, requiring only a traditional-style priority argument, namely the decidability of the existential theory of the r.e. degrees $\langle \mathbf{R}, \leq, \leq', 0, 1\rangle$ in the language of partial order, Turing reducibility of the jumps (i.e. $\mathbf{a} \leq' \mathbf{b}$ iff $\mathbf{a}' \leq \mathbf{b}'$), and least and greatest element.

Our notation is standard and follows Soare's book [20], especially its Chapter XIV, except that we define the use as the largest number *actually* used in a computation and denote the use function of a functional Φ^X by the corresponding lower-case letter φ (and similarly for other Greek letters).

2 The Theorems

We consider the r.e. degrees $\mathbf{R}$ in the language $\mathcal{L} = \{\leq, \leq', 0, 1, \}$ where the symbols denote in turn Turing reduciblity, Turing reducibility of the jumps (i.e. $\mathbf{a} \leq' \mathbf{b}$ iff $\mathbf{a}' \leq \mathbf{b}'$), and least and greatest element.

The main result of this paper is the following

Theorem 1. *The existential theory of* $\langle \mathbf{R}, \leq, \leq', 0, 1\rangle$ *is decidable.*

We prove this result by showing that all existential sentences not trivially false are actually true. The trivial restrictions are that $\leq$ and $\leq'$ are partial orderings, that $\leq$ is a sub-partial ordering of $\leq'$, and that 0 and 1 denote the (distinct) least and greatest elements, respectively, in both partial orderings.

More precisely, we use the following

Definition. *A* jump poset *is a 5-tuple* $\langle P, \leq, P', \leq', f\rangle$ *such that* $\langle P, \leq\rangle$ *and* $\langle P', \leq'\rangle$ *are partially ordered sets with (distinct) least and greatest element and* f *is a poset homomorphism from* P onto P' *preserving least and greatest elements.*

It is now easy to see that the proof of Theorem 1 can be reduced to proving the following

Theorem 2. *Any finite jump poset* $\langle P, \leq, P', \leq', f\rangle$ *can be embedded by maps* e *and* e' *in the following way:* e *is a poset embedding of* P *into* $\mathbf{R}$*,* e' *is a* *poset embedding of P' into the degrees REA in $\mathbf{0}'$ (i.e., r.e. in and above $\mathbf{0}'$), both e and e' preserve 0 and 1, and for all $p \in P$, $e(p)' = e'(f(p))$.*

3 Intuition for the Proof (of Theorem 2)

Our first inclination would be to first embed P' into the degrees REA in $\mathbf{0}'$ and then to invert the jump to construct an embedding of P. This works in some cases, for example if P is a linearly ordered set; but the Shore Noninversion Theorem [18] provides an easy counterexample: It is not always possible to invert the jump on an embedding of $P' = \{0, a, b, c, 1\}$ where $a, b < c$ and $a \mid b$, if $P = P'$ and f is the identity.

Thus we have to build an embedding of P into the r.e. degrees and control the jumps so as to ensure an embedding of P'. To simplify our notation, we will from now on denote the r.e. set of degree $e(p)$ (for $p \in P$) by A_p. Furthermore, without loss of generality (by possibly enlarging P), we can assume the following two conditions:

$$|P| \geq 4, \quad \text{and} \tag{1}$$

$$\exists p \in P \forall q \in P (q \neq 0, 1, p \rightarrow f(p) \mid' f(q)). \tag{2}$$

(These conditions will reduce the number of different strategies needed.)

We now set $A_0 = \emptyset$ and $A_1 = \emptyset'$. We will have to satisfy four types of requirements (for $p, q \in P$):

1. $p \leq q$ implies $A_p \leq_{\mathrm{T}} A_q$: This is only necessary for $p \neq 0$ and $q \neq 1$. In that case, any number x targeted for A_p will be greater than any other number used up to the time when x is picked, and when x later enters A_p it will simultaneously enter A_q. This strategy ensures $A_p \leq_{\mathrm{T}} A_q$. (Technically, we achieve $A_p = A_q \cap R$ for a recursive set R.)

2. $q \not\leq p$ implies $A_q \not\leq_{\mathrm{T}} A_p$. By (1) and (2), it suffices to ensure this only for $\{p, q\} \cap \{0, 1\} = \emptyset$. We use the strategy introduced by Friedberg [5] and Mučnik [15] to ensure $A_q \neq \Phi^{A_p}$ for each partial recursive (p.r.) functional Φ. Namely, we pick a (big) witness x, wait for $\Phi^{A_p}(x)\downarrow = 0$, and when and if that happens, we put x into A_q and restrain $A_p|(\varphi(x)+1)$.

For the other two types of requirements, we will use the Limit Lemma, which states that $X \leq_{\mathrm{T}} A'$ iff $X = \lim_s X_s$ for a uniformly A-recursive sequence $\{X_s\}_{s\in\omega}$.

3. $f(q) \not\leq' f(p)$ implies $A'_q \not\leq_{\mathrm{T}} A'_p$: By (1) and (2), it suffices to ensure this only for $\{p,q\} \cap \{0,1\} = \emptyset$. We use the strategy introduced by Lempp and Slaman [12] in their solution to the deep degree problem, constructing a p.r. functional Δ such that $\lim_s \Delta^{A_q}(-,s) \neq \lim_t \Psi^{A_p}(-,t)$ for any p.r. functional Ψ. First of all, we have to ensure that Δ^{A_q} is total and that $\lim_s \Delta^{A_q}(x,s)$ exists for all x. Furthermore, given Ψ, we pick a (big) witness x and start setting $\Delta^{A_q}(x,s) = 0$ for larger and larger s with some fixed large use u. Whenever we find a (new) t such that $\Psi^{A_p}(x,t){\downarrow}= 0$ then we restrain $A_p|(\psi(x,t)+1)$, put u into A_q, reset $\Delta^{A_q}(x,s) = 1$ for the "old" s (for which $\Delta^{A_q}(x,s)$ had been defined before), and keep setting $\Delta^{A_q}(x,s) = 0$ for the new s with new, even larger fixed use. In the absence of any injury, we thus achieve that
$$\exists^\infty t(\Psi^{A_p}(x,t) = 0) \to \forall s(\Delta^{A_q}(x,s) = 1), \quad \text{and} \tag{3.1}$$
$$\exists^{<\infty} t(\Psi^{A_p}(x,t) = 0) \to \text{a.e. } s(\Delta^{A_q}(x,s) = 0)\,. \tag{3.2}$$
(We remark here that this strategy makes A_p low_2.[1])

4. $f(p) \leq' f(q)$ implies $A'_p \leq_{\mathrm{T}} A'_q$: We distinguish two cases:

   4.1 $f(p) <' 1'$. In this case, we build a p.r. functional Γ ensuring $A'_p = \lim_s \Gamma^{A_q}(-,s)$. For each x, we start setting $\Gamma^{A_q}(x,s) = 0$ for larger and larger s until we find a stage s at which $\{x\}^{A_p}(x){\downarrow}$. Then we restrain $A_p|(u(A_p;x,x)+1)$ and start setting $\Gamma^{A_q}(x,s) = 1$ with some fixed large use u.

   4.2 $f(p) = 1'$. In this case, we make A_q high by building a p.r. functional Γ ensuring $\mathrm{Inf} = \lim_s \Gamma^{A_q}(-,s)$ where $\mathrm{Inf} = \{x \mid W_x \text{ infinite}\}$ is the canonical Π_2-complete set. For each x, we start setting $\Gamma^{A_q}(x,s) = 0$ for larger and larger s with some fixed large use u. Whenever a new element enters W_x we put u into A_q, reset $\Gamma^{A_q}(x,s) = 1$ for the "old" s, and keep setting $\Gamma^{A_q}(x,s) = 0$ for the "new" s with new, even larger fixed use. In the absence of any injury, we thus achieve that

[1] The authors would like to thank the referee for pointing this out.

$$x \in \text{ Inf } \to \forall s(\Gamma^{A_q}(x,s)=1), \text{ and} \tag{4.1}$$
$$x \notin \text{ Inf } \to \text{ a.e. } s(\Gamma^{A_q}(x,s)=0)\,. \tag{4.2}$$

When putting these strategies on a tree of strategies, there are three types of conflicts between strategies that go beyond conventional infinite-injury tree constructions:

A. Let α and β be two strategies where β is of lower priority than α and guesses that the restraint imposed by α tends to infinity. This can only be if α works on a requirement $A'_q \not\leq_{\mathrm{T}} A'_p$, and β may want to put numbers into a set $A_r \leq_{\mathrm{T}} A_p$ while α wants to restrain A_p and thus also A_r. We resolve this conflict by allowing β to injure α "in a controlled way", namely so that possibly infinitely many computations $\Psi^{A_p}(x,t)=0$ that α wants to preserve are injured but we ensure at the same time that infinitely many of these are preserved. This is achieved formally by "deactivating" β whenever α needs to find more permanent computations $\Psi^{A_p}(x,t)=0$ (which must happen under β's guess on α) and "activating" β when it would be allowed to injure a computation $\Psi^{A_p}(x,t)=0$ found by α now. If β's guess about α is correct, β will be activated infinitely often and still be able to satisfy its requirement.

B. Let α and β be two strategies where β is of lower priority than α and guesses that α puts infinitely many numbers into a set. Then α must work on a requirement $A'_q \not\leq_{\mathrm{T}} A'_p$ or Inf $\leq_{\mathrm{T}} A'_q$, and β may want to restrain a set $A_r \geq_{\mathrm{T}} A_q$ while α wants to put numbers into A_q and thus also into A_r. Notice that α puts an increasing sequence of numbers into A_q and that β knows, at any given stage, the lower bound for any future numbers put into A_q by α. We thus resolve this conflict by "delaying" β whenever it would restrain a set $A_r \geq_{\mathrm{T}} A_q$ above the current lower bound for any future numbers put into A_q by α. Since β wants to restrain to preserve some specific computation it assumes to be defined, the use of that computation must eventually be below the lower bound for any future numbers put into A_q by α (unless the computation is undefined) if β's guess about α is correct. (This feature is actually only necessary when combined with the next feature.)

   (The above two conflicts already appeared in Lempp and Slaman [12]; the third type of conflict is new and more complicated to handle.)

C. Fix a requirement of the form $A'_p(x) = \lim_s \Gamma^{A_q}(x,s)$. Since Γ's definition must be uniform, several strategies on the tree (in fact a maximal antichain of such strategies) all define $\Gamma^{A_q}(x,-)$ for the same x. It is conceivable that computations $\{x\}^{A_p}(x)$ (which would make us change $\Gamma^{A_q}(x,-)$ from 0 to 1) only appear when the $\Gamma^{A_q}(x,-)$-strategy whose guess currently seems correct is really to the right of the true path, so that its restraint to preserve $\{x\}^{A_p}(x)$ may later be injured while its definition $\Gamma^{A_q}(x,s) = 1$ may remain. If this happens infinitely often then $\Gamma^{A_q}(x,s) = 1$ for infinitely many s while $\{x\}^{A_p}(x)\uparrow$. To get around this problem, we introduce the "leftmost initialization feature". It consists of letting the leftmost $\Gamma^{A_q}(x,-)$-strategy β that is currently not delayed act even if a different $\Gamma^{A_q}(x,-)$-strategy $\hat{\beta} \neq \beta$ currently has the correct guess. This way, $\hat{\beta}$ can have β act in its place with β's higher-priority restraint to protect $\{x\}^{A_p}(x)$.

We are now in a position to present the formal proof of Theorem 2.

4 The Construction

We first define the requirements, keeping in mind the simplifying restrictions (1) and (2) above. Let

$$\begin{aligned}
Z_{0,0} &= \{\langle p,q\rangle \in (P-\{0,1\})^2 \mid q \not\leq p \ \ \& \ \ f(q) \leq' f(p)\}\,,\\
Z_{0,1} &= \{\langle p,q\rangle \in (P-\{0,1\})^2 \mid f(q) \not\leq' f(p)\}\,,\\
Z_1 &= \{\langle p,q\rangle \in P^2 \mid p \not\leq q \ \ \& \ \ f(p) \leq' f(q) \ \ \& \ \ f(p) \neq 1'\}\,,\\
Z_2 &= \{\langle 1,q\rangle \in P^2 \mid q \neq 1 \ \ \& \ \ f(q) = 1'\}\,.
\end{aligned}$$

For each $\langle p,q\rangle \in Z_{0,0}$ and each p.r. functional Φ, we require

$$A_q \neq \Phi^{A_p}\,. \tag{5}$$

For each $\langle p,q\rangle \in Z_{0,1}$, we build a p.r. functional $\Delta = \Delta_{\langle p,q\rangle}$ and require for each p.r. functional Ψ

$$\lim_s \Delta^{A_q}(-,s) \neq \lim_t \Psi^{A_p}(-,t)\,. \tag{6}$$

For each $\langle p,q\rangle \in Z_1$, we build a p.r. functional $\Gamma = \Gamma_{\langle p,q\rangle}$ and require for each x

$$\lim_s \Gamma^{A_q}(x,s) = A'_p(x)\,. \tag{7}$$

For each $\langle 1, q\rangle \in Z_2$, we build a p.r. functional $\Gamma = \Gamma_{\langle 1,q\rangle}$ and require for each x

$$\lim_s \Gamma^{A_q}(x,s) = \text{ Inf}(x)\,. \tag{8}$$

Finally recall that we automatically ensure for each $\langle p, q\rangle \in P^2$ with $p < q$

$$\exists \text{ a recursive set } R(A_p = A_q \cap R)\,. \tag{9}$$

Conditions (5) through (9) now ensure the embedding in Theorem 2.

We effectively ω-order all *requirements* (5) through (8) (for all Φ, Ψ, and x, respectively) as $\{\mathcal{R}_i\}_{i\in\omega}$. We let $T = 2^{<\omega}$, the full binary tree, be the *tree of strategies*, ordered lexicographically, and *assign* requirement $\mathcal{R}_i$ to all nodes $\sigma \in T$ of length i. We call the nodes σ of T *strategies*, and say σ *works for requirement* $\mathcal{R}_{|\sigma|}$. For each σ working on a requirement (5), (6), or (8), we define a *set of target sets* T_σ as follows:

If σ works on a requirement (5) then

$$T_\sigma = \{A_r \mid r \geq q\}\,.$$

If σ works on a requirement (6) then

$$T_\sigma = \{A_r \mid f(r) \geq' f(q)\}\,.$$

If σ works on a requirement (8) then

$$T_\sigma = \{A_r \mid f(r) = 1'\}\,.$$

(T_σ is undefined for σ working on a requirement (7).) The set of target sets is the collection of sets A_r into which σ will put numbers.

From time to time, strategies of T may be *initialized* by letting all their parameters (witness, restraint, etc.) be undefined. Note, however, that the functionals $\Gamma_{\langle p,q\rangle}$ and $\Delta_{\langle p,q\rangle}$ are global and thus never discarded. When a strategy σ is eligible to act it is *allowed to act* only if it is activated and not delayed. A strategy σ working on a requirement (5) or (6) is *deactivated* upon each initialization. σ becomes *activated* when its *priority parameter* $P(\sigma)$ (defined in the construction) is defined and less than the *counting parameter* i_τ (also defined in the construction) of any τ with $\tau\,\hat{}\,\langle 0\rangle \subseteq \sigma$ working on a requirement (6). (A strategy working on a requirement (7) or (8) is *always* activated.)

A strategy working on a requirement (5) through (7) is *delayed* if it would restrain a set $A_p \in T_\tau$ for some strategy τ working on a requirement (6) or (8) for which $\tau\hat{\ }\langle 0\rangle \subseteq \sigma$ and the A_p-restraint would be at least as big as τ's parameter u_τ (defined in the construction). (A strategy working on a requirement (8) is *never* delayed.)

A parameter is *defined big* by setting it equal to a number larger than any number used thus far in the construction.

The rest of this section now describes the action at each stage of the construction.

At stage 0, all strategies are initialized.

Each subsequent stage, $s+1$, say, has substages $t \leq s$. (Possibly, stage $s+1$ may be ended before reaching substage s.) At each substage t, a strategy σ of length t is *eligible to act* and then determines a strategy $\tau \supset \sigma$ to be eligible to act next (unless $t = s$ or the stage is ended).

At substage t of stage $s+1$, we first check if the strategy σ eligible to act is allowed to act. If not then define $P(\sigma)$ big (if now undefined) and end the substage by letting $\sigma\hat{\ }\langle 1\rangle$ be eligible to act next.

If σ is deactivated or delayed then we end the substage by letting $\sigma\hat{\ }\langle 1\rangle$ be eligible to act next; otherwise, we distinguish cases depending on which requirement σ works for:

Case 1: σ works on a requirement (5). If σ's witness x is undefined then define it big; in that case, or if σ's witness x is already in A_q or not $\Phi^{A_p}(x){\downarrow}= 0$, end the substage by letting $\sigma\hat{\ }\langle 1\rangle$ be eligible to act next. Otherwise, i.e. if $\Phi^{A_p}(x){\downarrow}= 0 = A_q(x)$, put x into every set $A_r \in T_\sigma$; restrain $A_p|(\varphi(x)+1)$, initialize all strategies $\tau > \sigma$, and end the substage by letting $\sigma\hat{\ }\langle 1\rangle$ be eligible to act next.

Case 2: σ works on a requirement (6). If σ's witness x is undefined, then define it big; for all $s' \leq s$ for which $\Delta^{A_q}(x,s'){\uparrow}$, set $\Delta^{A_q}(x,s') = 0$ with fixed big use $\delta(x,s') = u_\sigma$, say; set the counting parameter $i_\sigma = 0$; and end the substage by letting $\sigma\hat{\ }\langle 1\rangle$ be eligible to act next.

Otherwise, check if any of σ's A_p-restraint was injured. If so then let i_σ take the value it had just before σ imposed the least injured A_p-restraint. (Recall that i_σ counts the number of uninjured computations $\Psi^{A_p}(x,t){\downarrow}= 0$ that σ is currently preserving.) Next, check if there is a $t > i_\sigma$ such that $\Psi^{A_p}(x,t){\downarrow}= 0$. If not then, for all $s' \leq s$ for which $\Delta^{A_q}(x,s'){\uparrow}$, set $\Delta^{A_q}(x,s') = 1$ with previous use (if previously defined to equal 1), or set it

$= 0$ with use u_σ (where u_σ is the same as before if $u_\sigma \notin A_q$, and is chosen big otherwise); and end the substage by letting $\sigma\,\hat{}\,\langle 1\rangle$ be eligible to act next. Otherwise, i.e. if there is such a t, then restrain $A_p|(\psi(x,t)+1)$; put u_σ into every $A_r \in T_\sigma$; for all $s' \leq s$ for which $\Delta^{A_q}(x,s')\uparrow$, set $\Delta^{A_q}(x,s') = 1$ with previous use (if previously defined) or with use u_σ (otherwise); make the priority parameter $P(\sigma)$ undefined; increment the counting parameter i_σ by $+1$; and end the substage by letting $\sigma\,\hat{}\,\langle 0\rangle$ be eligible to act next.

Case 3: σ works on a requirement (7). Check if $\{x\}^{A_p}(x)\downarrow$. If not then set $\Gamma^{A_q}(x,s') = 0$ with use 0 for all $s' \leq s$ for which $\Gamma^{A_q}(x,s')$ is undefined and end the substage by letting $\sigma\,\hat{}\,\langle 1\rangle$ be eligible to act next. Otherwise, i.e. if $\{x\}^{A_p}(x)\downarrow$, then let τ be the leftmost strategy with $|\tau| = |\sigma|$ that has been eligible to act (since the last injury to a computation $\{x\}^{A_p}(x)$) and is currently not delayed. If this is the first time that τ acts for this computation $\{x\}^{A_p}(x)$ (since the last injury to a computation $\{x\}^{A_p}(x)$) then τ restrains $A_p|(u(A_p;x,x)+1)$; sets $\Gamma^{A_q}(x,s') = 0$ with use 0 for all $s' < s$ (for which $\Gamma^{A_q}(x,s')$ is undefined); sets $\Gamma^{A_q}(x,s) = 1$ with big use; initializes all strategies $\rho > \tau$; and ends the stage. Otherwise, σ sets $\Gamma^{A_q}(x,s')$ to the previous value with previous use (for $s' < s$); sets $\Gamma^{A_q}(x,s) = 1$ with big use; and ends the substage by letting $\sigma\,\hat{}\,\langle 1\rangle$ be eligible to act next.

Case 4: σ works on a requirement (8). If i_σ is undefined then define it big. If u_σ is undefined then define it big. Check if there is an element $y > i_\sigma$ in W_x. If not then set $\Gamma^{A_q}(x,s')$ (for $s' < s$) as previously defined with previous use; set $\Gamma^{A_q}(x,s) = 0$ with use $\gamma(x,s) = u_\sigma$; and end the substage by letting $\sigma\,\hat{}\,\langle 1\rangle$ be eligible to act next. Otherwise, i.e. if there is such a y, put u_σ into every $A_r \in T_\sigma$; for all $s' \leq s$ for which $\Gamma^{A_q}(x,s')$ is undefined set $\Gamma^{A_q}(x,s') = 1$ with use 0; redefine u_σ big; increment the counting parameter by $+1$; and end the substage by letting $\sigma\,\hat{}\,\langle 0\rangle$ be eligible to act next.

At the end of stage $s+1$, i.e. after all substages, initialize all strategies $\tau >$ the strategy σ that was last eligible to act; and define $\Delta^{A_q}(x,s')$ for all $x, s' \leq s$ for which it is now undefined as follows: If $\Delta^{A_q}(x,s')$ was previously defined then redefine it to the same value with the same use. Otherwise, if there is a (greatest) $s'' < s'$ such that $\Delta^{A_q}(x,s'')$ was previously defined then define $\Delta^{A_q}(x,s')$ to the same value with the same use as for s''. Otherwise, i.e., if there is no such s'', define it equal to 0 with use 0.

This ends the description of our construction for Theorem 2.

5 The Verification

We define the true path $f \in [T]$ by induction as usual, namely if $\sigma = f|n$ then $f(n) = 0$ if $\sigma\hat{}\langle 0\rangle$ is eligible to act infinitely often, and $f(n) = 1$ otherwise. We first prove a basic lemma:

Lemma 1 (True Path Lemma). *For each n, $f|n$ is initialized at most finitely often, is eligible to act infinitely often, and is activated at infinitely many of the stages at which it is eligible to act. Furthermore, if $\sigma = f|n$ is a strategy working on a requirement (6) or (8), then* $\lim_s i_\sigma$ *exists (possibly $= \infty$), and it is infinite iff $f(n) = 0$.*

Proof: We proceed by induction on n. The first half of the claim is trivial for $n = 0$, so we assume the claim for all $n' \leq n$ and set $\sigma = f|n$ in order to show the first half of the claim for $\tau = f|(n+1)$. First, suppose τ is initialized infinitely often. Then $\tau = \sigma\hat{}\langle 1\rangle$. There cannot be strategies $\tau' \supseteq \sigma\hat{}\langle 0\rangle$ working on a requirement (7) that cause initialization to τ infinitely often since there are only finitely many stages (and finitely many such τ') at which any τ' is eligible to act. Furthermore, the initialization of τ cannot be caused by $\sigma' \subset \sigma$ by the inductive hypothesis. Thus σ must initialize τ infinitely often. Since $\sigma\hat{}\langle 0\rangle \not\subset f$, σ must work on a requirement (7). Thus σ must be injured infinitely often, necessarily by strategies σ' with $\sigma'\hat{}\langle 0\rangle \subseteq \sigma$. These σ' must work on a requirement (6) or (8). But note that $u_{\sigma'}$ is a lower bound for any number σ' may put into any set in the future, and by the delay feature, σ will not try to restrain any set in $T_{\sigma'}$ on a number $\geq u_{\sigma'}$.

This also shows that τ must be eligible to act infinitely often since otherwise σ would end the stage almost every time it is eligible to act, thus initializing τ, which contradicts the above. Finally, suppose τ is activated only at finitely many of the stages at which it is eligible to act. Then $P(\tau)$ must come to a limit, but $\lim_s i_\rho = \infty$ for all ρ with $\rho\hat{}\langle 0\rangle \subset \tau$ working on a requirement (6) or (8), yielding the desired contradiction.

Finally, we need to show for all $\tau \subset f$ that, assuming the claim for all $\sigma \subset \tau$, if τ is a strategy working on a requirement (6) or (8) then $\lim_s i_\tau$ exists and is infinite iff $\tau\hat{}\langle 0\rangle \subset f$. Clearly $\lim_s i_\tau = \infty$ iff $\tau\hat{}\langle 0\rangle \subset f$ if $\lim_s i_\tau$ exists. Furthermore, $\lim_s i_\tau$ clearly exists if τ is a strategy working on a requirement (8) since then i_τ is nondecreasing in s. So assume τ is a strategy working on a requirement (6). It suffices to show for each fixed i, that if $i_\tau > i$ at infinitely many stages then $i_\tau > i$ at cofinitely many stages.

For the sake of a contradiction assume that each of $i_\tau > i$ and $i_\tau \leq i$ holds at infinitely many stages, respectively. By induction, we may assume $i_\tau \geq i$ at cofinitely many stages, say always after some (least) stage s, and say $s' > s$ is the least stage at which $i_\tau > i$. By the way we define $P(\rho)$, there are only finitely many strategies ρ at stage s' with $P(\rho) \leq i$. By the activation feature, only those ρ can cause $i_\tau \leq i$, and each can do so at most once and will then become deactivated. When $P(\rho)$ is redefined it will be defined big, and so this ρ can no longer injure the ith computation $\Psi^{A_p}(x,t)$ that τ finds. Thus eventually always $i_\tau > i$, yielding the desired contradiction. ⊠

Lemma 2 (Convergence Lemma). *All Γ^{A_q} and Δ^{A_q} are total, and $\lim_s \Delta^{A_q}(x,s)$ exists for all Δ and x.*

Proof: Since the use of $\Gamma^{A_q}(x,s)$ (or $\Delta^{A_q}(x,s)$), once defined, never increases and $\Gamma^{A_q}(x,s)$ (or $\Delta^{A_q}(x,s)$) is defined at the end of almost every stage, all Γ^{A_q} (and Δ^{A_q}) must be total.

Now fix Δ^{A_q} and x. We will show that $\lim_s \Delta^{A_q}(x,s)$ must exist. This is clear if almost all definitions of $\Delta^{A_q}(x,s)$ (for any s) are made at the end of stage s. Otherwise, a unique strategy σ makes definitions of $\Delta^{A_q}(x,s)$ (for some s) infinitely often, and necessarily $\sigma \subset f$. If $\sigma\hat{\ }\langle 1\rangle \subset f$ then clearly $\lim_s \Delta^{A_q}(x,s) = 0$. On the other hand, if $\sigma\hat{\ }\langle 0\rangle \subset f$ then, by σ's action at stages when it puts some u into A_q, $\Delta^{A_q}(x,s)$ will be set to 1 for almost all s. ⊠

Lemma 3 (Reducibility Lemma). *For all $p,q \in P$, if $p < q$ then $A_p \leq_{\mathrm{T}} A_q$.*

Proof: The claim is trivial for $p = 0$ or $q = 1$. Otherwise, any number x entering A_p must be appointed at a stage $s < x$. When x later enters A_p then it also enters A_q, which establishes the claim. ⊠

Lemma 4 (Non-Reducibility Lemma). *For any $p,q \in P$, if $q \not\leq p$ then $A_q \not\leq_{\mathrm{T}} A_p$.*

Proof: If $f(q) \not\leq' f(p)$ then we will show the (stronger) statement $A'_q \not\leq_{\mathrm{T}} A'_p$ in the next lemma. If $q = 1$ (or $p = 0$) then, by (1) and (2), we may pick an $r \in P - \{0,1\}$ with $r \not\leq p$ (or $q \not\leq r$), and the following proof will establish the (stronger) statement $A_r \not\leq_{\mathrm{T}} A_p$ (or $A_q \not\leq_{\mathrm{T}} A_r$, respectively). Thus we may assume $\langle p,q\rangle \in Z_{0,0}$ and, for each p.r. functional Φ, the existence of a strategy $\sigma \subset f$ working on $A_q \neq \Phi^{A_p}$.

Since σ is initialized at most finitely often, it will eventually work with a fixed witness x. So suppose $\Phi^{A_p}(x)\downarrow$. Then σ is not delayed at almost all stages since x and $\varphi(x)$ are eventually fixed but $\lim_s i_\tau = \infty$ for any τ with $\tau\,\hat{}\,\langle 0\rangle \subset \sigma$ working on a requirement (6) or (8) by Lemma 1. Thus clearly $\Phi^{A_p}(x) \neq 0$ if $x \notin A_q$.

So suppose σ puts x into A_q at some stage $s+1$. Then $\Phi^{A_p}(x)\downarrow= 0$ at that point, so suppose some $y \leq \varphi_{s+1}(x)$ is later put into A_p by some τ. By initialization by σ, we must have $\tau < \sigma$; and since σ keeps x forever and is thus not initialized after stage $s+1$, we even have $\tau\,\hat{}\,\langle 0\rangle \subseteq \sigma$. But then τ works on a requirement (6) or (8), so $\varphi_{s+1}(x) < u_{\tau,s+1}$. But any number τ puts into any set after stage $s+1$ must be $\geq u_{\tau,s+1}$ contradiction. Thus $\Phi^{A_p}(x)\downarrow= 0$ as desired. ⊠

Lemma 5 (Jump Non-Reducibility Lemma). *For any $p, q \in P$, if $f(q) \not\leq' f(p)$ then $A'_q \not\leq_{\mathrm{T}} A'_p$.*

Proof: If $q = 1$ (or $p = 0$) then, by (1) and (2), we may pick an $r \in P - \{0,1\}$ with $f(r) \not\leq' f(p)$ (or $f(q) \not\leq' f(r)$), and the following proof will establish the (stronger) statement $A'_r \not\leq_{\mathrm{T}} A'_p$ (or $A'_q \not\leq_{\mathrm{T}} A'_r$, respectively). Thus we may assume $\langle p, q\rangle \in Z_{0,1}$ and, for each p.r. functional Ψ, the existence of a strategy $\sigma \subset f$ working on $\lim_s \Delta^{A_q}(-, s) \neq \lim_t \Psi^{A_p}(-, t)$.

Since σ is initialized at most finitely often, it will eventually work with a fixed witness x. So suppose $\Psi^{A_p}(x, t)\downarrow= 0$ or 1 (for all t) and that $\lim_t \Psi^{A_p}(x, t)$ exists.

If $\sigma\,\hat{}\,\langle 1\rangle \subset f$, then, after stage s, say, i_σ is constant, say equal to i_0. Suppose there is a $t > i_0$ such that $\Psi^{A_p}(x, t)\downarrow= 0$. Then x and $\psi(x, t)$ are eventually fixed but $\lim_s i_\tau = \infty$ for any τ with $\tau\,\hat{}\,\langle 0\rangle \subseteq \sigma$ working on a requirement (6) or (8) by Lemma 1; thus i_σ would eventually be greater than i_0, a contradiction. So $\sigma\,\hat{}\,\langle 1\rangle \subset f$ implies not $\lim_t \Psi^{A_p}(x, t) = 0$, and, by the proof of Lemma 2, it also implies $\lim_s \Delta^{A_q}(x, s) = 0$.

If $\sigma\,\hat{}\,\langle 0\rangle \subset f$ then $\lim_s i_\sigma = \infty$; so by the injury feature, there are infinitely many t such that $\Psi^{A_p}(x, t)\downarrow= 0$. Thus $\sigma\,\hat{}\,\langle 0\rangle \subset f$ implies not $\lim_t \Psi^{A_p}(x, t) = 1$, and, again by the proof of Lemma 2, it also implies $\lim_s \Delta^{A_q}(x, s) = 1$. ⊠

Lemma 6 (Jump Reducibility Lemma). *For any $p, q \in P$, if $f(p) \leq' f(q)$ then $A'_p \leq_{\mathrm{T}} A'_q$.*

Proof: If $p \leq q$ then we have already established the (stronger) statement $A_p \leq_{\mathrm{T}} A_q$ by Lemma 3. The claim is trivial for $f(p) = 0'$ or $f(q) = 1'$. So we may assume $p \not\leq q$, $f(p) \neq 0'$, and $f(q) \neq 1'$. We now distinguish two cases:
Case 1: $f(p) \neq 1'$. Then $\langle p, q\rangle \in Z_1$, and for each x there is a strategy $\sigma \subset f$ working on $\lim_s \Gamma^{A_q}(x, s) = A'_p(x)$. Let σ not be initialized after a (least) stage s_0. If σ (or some $\sigma' <_L \sigma$ with $|\sigma| = |\sigma'|$) finds a computation $\{x\}^{A_p}(x)\!\downarrow$ (and is not initialized afterwards) while it is not delayed then it will initialize all strategies $\tau > \sigma$ (or $\tau > \sigma'$, respectively). By the delay feature and initialization, that computation $\{x\}^{A_p}(x)\!\downarrow$ will not be injured, and by the construction $\Gamma^{A_q}(x, s) = 1$ for almost all s.

On the other hand, if σ (or some $\sigma' <_L \sigma$ with $|\sigma| = |\sigma'|$) never finds such a computation then $\{x\}^{A_p}(x)\downarrow$ is impossible since then $u(A_p; x, x)$ would eventually be fixed while $\lim_s u_\tau = \infty$ for all τ working on a requirement (6) or (8) with $\tau\,\hat{}\,\langle 0\rangle \subseteq \sigma$. So suppose some strategy $\overline{\sigma} >_{\mathrm{L}} \sigma$ sets $\Gamma^{A_q}(x, s) = 1$ at a stage $s > s_0$ while σ is delayed. Then σ is delayed because $u_\tau \leq u(A_p; x, x, s)$ for some τ working on a requirement (6) or (8) with $\tau\,\hat{}\,\langle 0\rangle \subseteq \sigma$. Since $\tau\,\hat{}\,\langle 0\rangle \subseteq \sigma \subset f$, τ will later put u_τ into A_p and thus also into A_q (by $f(q) \geq' f(p)$). Since $\gamma(x, s) > u(A_p; x, x, s)$, this will destroy $\overline{\sigma}$'s definition of $\Gamma^{A_q}(x, s) = 1$. Thus $\Gamma^{A_q}(x, s) = 0$ for almost all s.

Case 2: $f(p) = 1'$. Then $\langle 1, q\rangle \in Z_2$, and for each x there is a strategy $\sigma \subset f$ working on $\lim_s \Gamma^{A_q}(x, s) = \mathrm{Inf}(x)$. Let σ not be initialized after a (least) stage s_0. By the construction, it is easy to see that $\Gamma^{A_q}(x, s) = 0$ for almost all s if W_x is finite.

On the other hand, if W_x is infinite then σ will infinitely often reset $\Gamma^{A_q}(x, -)$ from 0 to 1. Let s_1 be the first stage after stage s_0 at which σ is eligible to act. Then, for any $s' \geq s_1$ and at any stage $\geq s_1$, $\gamma(x, s') \geq u_\sigma$ or $\Gamma^{A_q}(x, s') = 1$; so whenever σ resets $\Gamma^{A_q}(x, -)$ we have $\Gamma^{A_q}(x, s') = 1$ for all s' with $s_1 \leq s' \leq$ the current stage. Thus $\Gamma^{A_q}(x, s) = 1$ for almost all s as desired. ⊠

The above lemmas conclude the proof of Theorem 2 and thus also of Theorem 1.

6 Final Remarks

The above construction contains three features distinguishing it from "usual" infinite-injury priority arguments: firstly the activation/deactivation feature; secondly the delay feature; and thirdly the leftmost initialization feature.

These features seem strangely ad hoc but they are actually quite natural in a more general framework such as the one developed by us for the general result, including higher jump reducibility predicates. The activation/deactivation feature, controlling the injury by a lower-priority strategy σ to a higher-priority strategy τ with $\tau\hat{\ }\langle 0\rangle \subseteq \sigma$, is merely a tool to assign relative priority to the substrategies of σ and τ, each performing a single action, in such a way that each substrategy has only finitely many substrategies (of both σ and τ) of higher priority. It reflects the priority ordering on T_1, the "finite-injury tree".

The delay feature, preventing a lower-priority strategy σ from being injured by a higher-priority strategy τ with $\tau\hat{\ }\langle 0\rangle \subseteq \sigma$, is a tool to assert the priority of τ (as a whole) over σ in spite of the controlled injury by σ-substrategies to τ. It reflects the priority ordering on T_2, the "infinite-injury tree".

Finally, the leftmost initialization feature, enabling a finitary strategy τ to act even while it is not currently on the true path, is a tool to "force τ back on the true path" since it may be too late for τ to act when it returns to the true path. It reflects forcing a strategy currently on the true path of T_1 to be put back on the true path of T_2. This feature will be a crucial component of our general general framework.

All three features should become clearer in our forthcoming paper [11], to be written in our new general framework for priority arguments.

References

[1] K. Ambos-Spies, M. Lerman, Lattice embeddings into the recursively enumerable degrees, *J. Symbolic Logic* **51** (1986), 257–272.

[2] K. Ambos-Spies, M. Lerman, Lattice embeddings into the recursively enumerable degrees, II, *J. Symbolic Logic* **54** (1989), 735–760.

[3] S. B. Cooper, The jump is definable in the structure of the degrees of unsolvability, preprint.

[4] R. L. Epstein, "Degrees of Unsolvability: Structure and Theory", Lecture Notes in Mathematics No. 759, Springer-Verlag, Berlin, Heidelberg, New York, 1979.

[5] R. M. Friedberg, Two recursively enumerable sets of incomparable degrees of unsolvability, *Proc. Natl. Acad. Sci. USA* **43** (1957), 236–238.

[6] L. Harrington, S. Shelah, The undecidability of the recursively enumerable degrees, *Bull. Amer. Math. Soc. (N. S.)* **6** No. 1 (1982), 79–80.

[7] L. Harrington, T. A. Slaman, Interpreting arithmetic in the Turing degrees of the recursively enumerable sets, to appear.

[8] P. G. Hinman, T. A. Slaman, Jump embeddings into the Turing degrees, *J. Symbolic Logic* **56** (1991), 563–591.

[9] A. H. Lachlan, Distributive initial segments of the degrees of unsolvability, *Z. math. Logik Grund. Math.* **14** (1968), 457–472.

[10] S. Lempp, M. Lerman, Priority arguments using iterated trees of strategies, in "Recursion Theory Week", K. Ambos-Spies, G.H. Müller, G.E. Sacks (eds.), Springer-Verlag, Berlin, 1990, pp. 277-296.

[11] S. Lempp, M. Lerman, The decidability of the existential theory of the poset of the recursively enumerable degrees with jump relations, in preparation.

[12] S. Lempp, T. A. Slaman, A limit on relative genericity in the recursively enumerable degrees, *J. Symbolic Logic* **54** (1989), 376–395.

[13] M. Lerman, "Degrees of unsolvability", Perspectives in Mathematical Logic, Omega Series, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1983.

[14] M. Lerman, R. A. Shore, Decidability and invariant classes for degree structures, *Trans. Amer. Math. Soc.* **301(2)** (1988), 669–692.

[15] A. M. Mučnik, On the solvability of the problem of reducibility in the theory of algorithms, *Dokl. Akad. Nauk SSSR, N.S.* **108** (1956), 194–197.

[16] R. A. Shore, On the ∀∃-sentences of α-recursion theory, in: "Generalized Recursion Theory II", J. E. Fenstad, R. O. Gandy, G. E. Sacks (eds.), Proceedings of the Second Symposium on Generalized Recursion Theory, Oslo, 1977, North Holland, Amsterdam, New York, Oxford, 1978, pp. 331–354.

[17] R. A. Shore, On homogeneity and definability in the first-order theory of the Turing degrees, *J. Symbolic Logic* **47** (1982), 8–16.

[18] R. A. Shore, A non-inversion theorem for the jump operator, *Ann. Pure Appl. Logic* **40** (1988), 277–303.

[19] S. G. Simpson, First-order theory of the degrees of recursive unsolvability, *Ann. Math.* **(2) 105** (1977), 121–139.

[20] R. I. Soare, "Recursively enumerable sets and degrees", Perspectives in Mathematical Logic, Omega Series, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1987.