A HIGH STRONGLY NONCAPPABLE DEGREE

STEFFEN LEMPP

ABSTRACT. An r.e. degree $\mathbf{a} \neq \mathbf{0}, \mathbf{0}'$ is called strongly noncappable if it has no inf with any incomparable r.e. degree. We show the existence of a high strongly noncappable degree.

0. Introduction. The early study of $\mathbf{R}$ revealed certain "nice" properties. For example, the Sacks Splitting Theorem [**Sa63**] showed that any nonrecursive r.e. degree is the supremum of two incomparable r.e. degrees. The Sacks Density Theorem [**Sa64**] showed that $\mathbf{R}$ is a dense partial order. These and other results led Shoenfield [**Sh65**] to conjecture that if $\vec{\mathbf{a}} \in \mathbf{R}$ satisfies a diagram $D(\vec{x})$ in the language $\mathcal{L} = \{0, 1, \leq, \cup\}$ of upper semilattices and $D_0(\vec{x}, y)$ is a consistent extension of $D(\vec{x})$, then there is $\mathbf{b} \in \mathbf{R}$ such that $\vec{\mathbf{a}}$ and $\mathbf{b}$ satisfy $D_0(\vec{x}, y)$. A consequence of this would be that no two incomparable r.e. degrees have an infimum (*cap* to some lower r.e. degree). This was refuted independently by Lachlan [**La66**] and Yates [**Ya66**] through the construction of a *minimal pair* (capping to $\mathbf{0}$). Yates [ibid.] also showed that some r.e. degrees are *noncappable* (not half of a minimal pair). Soare [**So80**] defined the notion of a *strongly noncappable* (*s.n.c.*) degree (an r.e. degree $\neq \mathbf{0}, \mathbf{0}'$ that does not have an infimum with any incomparable r.e. degree). Ambos-Spies [**AS84**] proved the existence of s.n.c. degrees and various stronger results, but all his such degrees were constructed by finite injury arguments and thus are low. The $\mathbf{0}'''$-priority argument in this paper establishes the existence of a strongly noncappable degree, which is high. This is a step in the characterization of the range of the *jump* operator on certain classes of r.e. degrees. Which degrees actually are the jumps of s.n.c. degrees still remains an open question. A recent related result by Cooper [**Cota**] (and independently by Shore [**Shta**]) shows that the range of the jump operator on the set of cappable degrees is not the set of all degrees r.e. in and above (REA in) $\mathbf{0}'$.

Our notation is fairly standard and generally follows Soare's forthcoming book "Recursively Enumerable Sets and Degrees" [**Sota**].

We consider sets and functions on the natural numbers $\omega = \{0, 1, 2, 3, \dots\}$. Usually lower-case Latin letters $a, b, c, \dots$ denote natural numbers; $f, g, h, \dots$ total functions on ω; Greek letters $\Phi, \Psi, \dots, \varphi, \psi, \dots$ partial functions on ω; and upper-case Latin letters $A, B, C, \dots$ subsets of ω. For a partial function φ, $\varphi(x) \downarrow$ denotes that $x \in \operatorname{dom} \varphi$, otherwise we write $\varphi(x) \uparrow$. We identify a set A with its characteristic function χ_A. $f \restriction x$ denotes f restricted to arguments less than x, likewise for sets.

We let $A \subset B$ denote that $A \subseteq B$ but $A \neq B$; and $A \subseteq^* B$ that $A - B$ is finite. $A \sqcup B$ will denote the disjoint union. For each $n \in \omega$, we let $\langle x_1, x_2, \dots, x_n \rangle$ denote the coded n-tuple (where $x_i \leq \langle x_1, x_2, \dots, x_n \rangle$ for each i); and $(x)_i$ the ith projection function, mapping $\langle x_1, x_2, \dots, x_n \rangle$ to x_i. $A^{[k]} = \{y \mid \langle y, k, \in \rangle A\}$ denotes the kth "row" of A.

This paper is part of the author's thesis. He wishes to thank his thesis advisor, R.I. Soare, as well as T.A. Slaman for many helpful suggestions and comments.

In a partial order, $x \mid y$ denotes that x and y are incomparable. The logical connectives "and" and "or" will be denoted by $\wedge$ and $\vee$, respectively. We allow as an additional quantifier (in the meta-language) $(\exists^\infty x)$ to denote that the set of such x is infinite.

$\{e\}$ (or φ_e) and W_e ($\{e\}^X$ (or Φ_e^X) and W_e^X) denote the eth partial recursive function and its domain (with oracle X) under some fixed standard numbering. $\leq_T$ denotes Turing reducibility, and $\equiv_T$ the induced equivalence relation. The *use* of a computation $\Phi_e^X(x)$ (denoted by $u(X;e,x)$) is 1 plus the largest number from oracle X used in the computation if $\Phi_e^X(x)\downarrow$; and 0 otherwise (likewise for $u(X;e,x,s)$, the use at stage s). Sets, functionals, and parameters are often viewed as being in a state of formation, so, when describing a construction, we may write A (instead of the full Lachlan notation A_s, $A[s]$, or $A_t[s]$ for the value at the end of stage s or at the end of substage t of stage s).

In the context of trees, $\rho, \sigma, \tau, \ldots$ denote *finite strings*; $|\sigma|$ the *length* of σ; $\sigma\widehat{\ }\tau$ the *concatenation* of σ and τ; $\langle a \rangle$ the one-element string consisting of a; $\sigma \subseteq \tau$ ($\sigma \subset \tau$) that σ is a (*proper*) *initial segment* of τ; $\sigma <_L \tau$ that for some i, $\sigma \restriction i = \tau \restriction i$ and $\sigma(i) <_\Lambda \tau(i)$ (where $<_\Lambda$ is a given order on Λ and $T \subseteq \Lambda^{<\omega}$); and $\sigma \leq \tau$ ($\sigma < \tau$) that $\sigma <_L \tau$ or $\sigma \subseteq \tau$ ($\sigma \subset \tau$). The set $[T]$ of *infinite paths* through a tree $T \subseteq \Lambda^{<\omega}$ is $\{ p \in \Lambda^\omega \mid (\forall n)[p \restriction n \in T] \}$.

We use the following conventions: Upper-case letters at the beginning of the alphabet are used for sets $A, B, C, \ldots$ and functionals $\Gamma, \Delta, \ldots$ constructed by *us*; those at the end of the alphabet are used for sets $U, V, W, \ldots$ and functionals $\Phi, \Psi, \ldots$ constructed by the *opponent*. A functional Φ ($\Psi, \Theta, \ldots$) is viewed as an r.e. set of triples $\langle x, y, \sigma \rangle$ (denoting $\Phi^\sigma(x)\downarrow = y$), and the corresponding Greek lower-case letter φ ($\psi, \vartheta, \ldots$) denotes a modified use function for Φ ($\Psi, \Theta, \ldots$), namely, $\varphi(x) = |\sigma| - 1$ (so changing X at $\varphi(x)$ will change $\Phi^X(x)$). Parameters, once assigned a value, retain this value until reassigned.

Strategies are identified with strings on the tree corresponding to their guess about the outcomes of higher-priority strategies and are viewed as finite automata described in flow charts. In these flow charts, states are denoted by circles, instructions to be executed by rectangles, and decisions to be made by diamonds. To *initialize* a strategy means to put it into state *init* and to set its restraint to zero. A strategy is initialized at stage 0 and whenever specified later. At a stage when a strategy is allowed to act, it will proceed to the next state along the arrows and according to whether the statements in the diamonds are true (y) or false (n). Along the way, it will execute the instructions. Half-circles denote points in the diagram where a strategy starts from through the action of another strategy. Sometimes, parts of a flow chart are shared, the arrows are then labeled i and ii. The *strategy control* decides which strategy can act when. For some further background on $\mathbf{0}'''$-priority arguments, we refer to Soare ([Sota] or [So85])

1. The Theorem. Soare [So80] defined:

DEFINITION: An r.e. degree $\mathbf{a} \neq \mathbf{0}, \mathbf{0}'$ is *strongly noncappable* (*s.n.c.*) if $\mathbf{a}$ does not have an infimum with any incomparable r.e. degree $\mathbf{v}$, i.e., in the r.e. degrees,

$$(1) \qquad (\forall \mathbf{v})(\forall \mathbf{u})\big[\mathbf{a} \mid \mathbf{v} \wedge \mathbf{u} \leq \mathbf{a}, \mathbf{v} \rightarrow (\exists \mathbf{b})[\mathbf{b} \leq \mathbf{a}, \mathbf{v} \wedge \mathbf{b} \not\leq \mathbf{u}]\big].$$

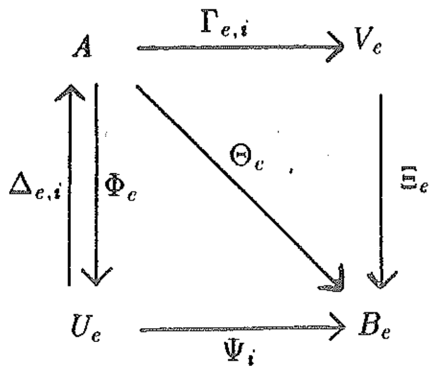

Diagram 1. Sets and functionals used

Ambos-Spies [AS84] showed the existence of various low s.n.c. degrees. We will show:

THEOREM. *There is a high strongly noncappable degree.*

PROOF: Actually, we will prove, similarly to Ambos-Spies, a slightly stronger result, namely, we will construct a high r.e. degree $\mathbf{a} \neq \mathbf{0}'$ such that in the r.e. degrees,

(2) $$(\forall \mathbf{v})(\forall \mathbf{u})[\mathbf{u} < \mathbf{a} \wedge \mathbf{v} \not\leq \mathbf{a} \rightarrow (\exists \mathbf{b})[\mathbf{b} \leq \mathbf{a}, \mathbf{v} \wedge \mathbf{b} \not\leq \mathbf{u}]].$$

(This implies (1) by letting $\mathbf{u} \leq \mathbf{v}$ also.)

2. The Requirements. We will build a high r.e. set A of s.n.c. degree by satisfying the following three requirements:

To ensure that A is high we let J be an r.e. set which in the limit codes $\emptyset''$ as follows:

(3) $$(\forall e)[(e \in \emptyset'' \rightarrow J^{[2e]} =^* \emptyset) \wedge (e \notin \emptyset'' \rightarrow J^{[2e]} = \omega^{[2e]})].$$

Then the usual thickness requirements will suffice to make A high:

(4) $$\mathcal{P}_e : A^{[2e]} =^* J^{[2e]}.$$

To make A incomplete we require for all e:

(5) $$\mathcal{N}_e : K \neq \{e\}^A,$$

where $K = \emptyset'$ (although we could in this construction replace K by any nonrecursive r.e. set W). Our basic strategy for $\mathcal{N}_e$ will be the Sacks preservation strategy, using a typical tree argument to deal with infinite injury from the $\mathcal{P}$-strategies but a new coding strategy for such injury from the $\mathcal{S}$-strategies as explained below.

To ensure (2) for strong noncappability, we stipulate that for all e,

(6) $$\tilde{\mathcal{R}}_e : U_e = \Phi_e^A \rightarrow [A \leq_T U_e \vee V_e \leq_T A \vee (\exists B_e)[B_e \leq_T A, V_e \wedge B_e \not\leq_T U_e]],$$

where $\{U_e, V_e, \Phi_e\}_{e \in \omega}$ is an enumeration of all triples of r.e. sets U, V and functionals Φ (given by the opponent), and where the B_e are built by us. (See Diagram 1.)

However, the $\tilde{\mathcal{R}}_e$ are still too complicated to be satisfied at one level of the tree, so we split each $\tilde{\mathcal{R}}_e$ up into

(7) $$\hat{\mathcal{R}}_e : U_e = \Phi_e^A \to B_e \leq_T A, V_e,$$

and for all $i \in \omega$,

(8) $$\hat{\mathcal{S}}_{e,i} : U_e = \Phi_e^A \wedge B_e = \Psi_i^{U_e} \to A \leq_T U_e \vee V_e \leq_T A,$$

where $\{ \Psi_i \}_{i \in \omega}$ is an enumeration of all functionals Ψ (given by the opponent).

For the sake of $\hat{\mathcal{R}}_e$, we will build functionals Θ_e, Ξ_e such that

(9) $$\mathcal{R}_e : U_e = \Phi_e^A \to B_e = \Theta_e^A \wedge B_e = \Xi_e^{V_e}.$$

For $\hat{\mathcal{S}}_{e,i}$, we will construct functionals $\Gamma_{e,i}, \Delta_{e,i}$ such that

(10) $$\mathcal{S}_{e,i} : U_e = \Phi_e^A \wedge B_e = \Psi_i^{U_e} \to A =^* \Delta_{e,i}^{U_e} \vee V_e =^* \Gamma_{e,i}^A.$$

The $\mathcal{R}_e$ and $\mathcal{S}_{e,i}$ will correspond to actual strategies.

The strategies for satisfying the requirements will be arranged on nodes of a tree. Each strategy will be responsible for one requirement of type $\mathcal{N}$, $\mathcal{P}$, $\mathcal{R}$, or $\mathcal{S}$ and will from now on be called $\mathcal{N}$-, $\mathcal{P}$-, $\mathcal{R}$-, or $\mathcal{S}$-strategy. (We will suppress indices whenever they are clear from the context.)

3. Making A S.N.C. and Incomplete. In order to be able to restrain U through A, we will require that

(11) $$x \in U_{s+1} - U_s \to \Phi^A(x)[s] = 1.$$

Then $\Phi^{A \restriction u}[s] \restriction x = U_s \restriction x$ and $A_s \restriction u = A \restriction u$ implies $U_s \restriction x = U \restriction x$. We also tacitly assume that all use functions $\varphi_s(x)$, etc. are increasing in x and nondecreasing in s.

For satisfying $\tilde{\mathcal{R}}_e$, we have to ensure first of all $\mathcal{R}_e$. Each $\mathcal{R}_e$-strategy α will build its version of Ξ_e as direct permitting on α-stages $(V_{e,s} \restriction x = V_e \restriction x \wedge s \in S^\alpha \to B_{e,s}(x) = B_e(x))$, and we will therefore not mention Ξ_e any more. However, V_e and Ξ_e are used by many strategies on the cone below the $\mathcal{R}_e$-strategy. Therefore, in our infinite injury setting, direct permitting requires that the strategy responsible for building Ξ_e (i.e., the $\mathcal{R}_e$-strategy) allow a strategy below on the tree to act immediately if the latter wants to put a number into B_e and thus needs a V_e-change to correct Ξ_e. A version of the functional Θ_e will be built explicitly by each $\mathcal{R}_e$-strategy as the length of agreement between U and Φ_e^A increases. Notice thus that an $\mathcal{R}$-strategy only builds a functional, but does not enumerate numbers into any set or impose any restraint. Its outcomes are $\Phi^A \neq U$ (called 1, in which case Θ will be finite), and (a guess that) $\Phi^A = U$ (called 0, in which case it has to ensure that Θ^A is total and $\Theta^A = B$).

An $\mathcal{S}_{e,i}$-strategy β, which will only ever act if it is below the outcome 0 of an $\mathcal{R}_e$-strategy on the tree, will mainly try to "code V_e into A" by gradually building $\Gamma_{e,i}$ and putting $\gamma_{e,i}(x)$ into A whenever $\Gamma^A_{e,i}(x) \downarrow \neq V_e(x)$ (to ensure the correctness of $\Gamma_{e,i}$). If $V_e = K$ then this would make A complete and thus injure one of the $\mathcal{N}$-strategies below, say, $\gamma \supset \beta$. So the *key to the whole construction* is the feature that the $\mathcal{N}$-strategy γ helps the $\mathcal{S}_{e,i}$-strategy β prove $B_e \neq \Psi_i^{U_e}$ and then *immediately* shuts β off. The outcomes of the $\mathcal{S}_{e,i}$-strategy β are again 0 (infinite action) and 1 (finite action).

Now consider an $\mathcal{N}_e$-strategy γ, and assume it is on the true path and thus has to satisfy its requirement. The strategies to the left of γ only have finite effect; γ will put up restraint against the strategies to the right of and below γ. So the only strategies dangerous to γ lie above it on the tree, and they are either $\mathcal{P}_{e'}$ or $\mathcal{S}$-strategies. The former are no problem since γ knows their outcome (either $A^{[2e']} =^* \omega^{[2e']}$ or $A^{[2e']} =^* \emptyset$). For each $\mathcal{S}$-strategy $\beta \subset \gamma$ for which γ guesses that β puts infinitely many numbers into A, γ will try to take over β's responsibility and to put up a candidate x for $B(x) \neq \Psi^U(x)$.

If γ succeeds in finding a suitable candidate, there are two possibilities: Either V will change and allow x into B, while the $\mathcal{N}$-strategy preserves $\Psi^U(x) = 0$; thus $B(x) = 1 \neq 0 = \Psi^U(x)$. Then β's requirement has been satisfied by γ, therefore β can be shut off and has finite outcome. So γ is not on the true path after all, and its restraint will have the same priority as if it were imposed by β (since no $\xi \supseteq \beta^\frown\langle 0 \rangle$ will act ever again unless β is initialized). The other possibility is that V does not change, which constitutes another step towards showing that $V \leq_T A$.

The strategy γ may have to act even when it is not its turn since it needs to redefine a functional of much higher priority. Thus γ might injure higher-priority strategies which have increased their restraint since γ acted last. Therefore, whenever some $\mathcal{N}$-strategy γ' changes states (while it is its turn), the strategy control will initialize all strategies $\xi > \gamma'$ to prevent them from injuring γ'. This is compatible with the rest of the construction since each $\mathcal{N}$-strategy γ on or to the left of the true path will act only finitely often.

On the other hand, if γ fails to find a suitable candidate, then β has to make Δ total and ensure that $\Delta^U = A$. So again β's requirement will be ensured by γ.

Candidates x for showing $B \neq \Psi^U$ must have the property that $\vartheta(x) > \varphi(\psi(x))$ so that we can put x into B, put $\vartheta(x)$ into A to correct $\Theta^A(x)$, and at the same time restrain $A \upharpoonright (\varphi(\psi(x)) + 1)$ to preserve $U \upharpoonright (\psi(x) + 1)$ and thus $\Psi^U(x) = 0$. Now an $\mathcal{R}$-strategy can wait with the definition of $\Theta^A(x)$ until $\Phi^A \upharpoonright (y + 1)$ is defined (for some y depending on x), but not for $\Psi^U(x)$ (which may not be defined at all). So we introduce the A-recursive *computation function* of A,

$$c_A(x) = \mu s[A_s \upharpoonright (x+1) = A \upharpoonright (x+1)]$$

for the given enumeration of A, and its recursive approximation

$$c_A(x, s) = (\mu t \leq s)[A_s \upharpoonright (x+1) = A_t \upharpoonright (x+1)].$$

Now if $U <_T A$, ψ is a U-recursive function, and S is an infinite U-recursive set then

$$(\exists^\infty x \in S)[\psi(x) < c^A(x)],$$

and thus if in addition $\Phi^A = U$ and φ is increasing then

$$(\exists^\infty x \in S)[\varphi(\psi(x)) < \varphi(c^A(x))].$$

If $U <_T A$, this will ensure that an $\mathcal{N}$-strategy below an $\mathcal{S}$-strategy can find enough candidates x for $B(x) \neq \Psi^U(x)$ with $\vartheta(x) > \varphi(\psi(x))$ by having at stage $s+1$ the $\mathcal{R}$-strategy put $\vartheta(x) > \varphi(c_A(x,s))$. (The function c_A is Ambos-Spies's function γ as explained in Lemma 1 of [AS84].) On the other hand, if an $\mathcal{N}$-strategy γ cannot find a suitable candidate for an $\mathcal{S}$-strategy $\beta \subset \gamma$, we can allow γ to shut off β *eventually*.

The outcome of the $\mathcal{N}$-strategy γ is the lim inf of the restraint that γ imposes on the lower-priority strategies. Note that only the $\mathcal{N}$-strategies want to restrain A.

4. The Full Construction. We will first describe the tree of strategies and then give the full module for each type of strategy (in a flow chart) and explain the strategy control to see how the strategies interact.

Let $\Lambda_\mathcal{N}$, $\Lambda_\mathcal{P}$, $\Lambda_\mathcal{R}$, and $\Lambda_\mathcal{S}$ be the *sets of outcomes* of the $\mathcal{N}$-, $\mathcal{P}$-, $\mathcal{R}$-, and $\mathcal{S}$-strategies (where $\Lambda_\mathcal{N} = \omega$ and $\Lambda_\mathcal{P} = \Lambda_\mathcal{R} = \Lambda_\mathcal{S} = \{0,1\}$), and let Λ be their union. The *tree of strategies* is

(12) $$T = \{\, \xi \in \Lambda^{<\omega} \mid (\forall k < |\xi|)[\xi(k) \in \Lambda_\mathcal{N}, \Lambda_\mathcal{P}, \Lambda_\mathcal{R}, \Lambda_\mathcal{S} \text{ for } k \equiv 0,1,2,3 \bmod 4] \,\}.$$

To each node $\xi \in T$, we assign a type of strategy ($\mathcal{N}$, $\mathcal{P}$, $\mathcal{R}$, $\mathcal{S}$ for $|\xi| \equiv 0,1,2,3 \bmod 4$) and a number $e(\xi)$ (or $\langle e(\xi), i(\xi) \rangle) = \frac{|\xi|-k}{4}$ (for some $k \in \{0,1,2,3\}$) so that ξ works on requirement $\mathcal{N}_{e(\xi)}$, $\mathcal{P}_{e(\xi)}$, $\mathcal{R}_{e(\xi)}$, or $\mathcal{S}_{e(\xi),i(\xi)}$. Then for each infinite path $h \in [T]$, there is exactly one strategy $\xi \subset h$ working on each requirement. Fixing e and i, notice that if α is the $\mathcal{R}_e$-strategy $\alpha \subset h$ and β is the $\mathcal{S}_{e,i}$-strategy $\beta \subset h$, we have that $\alpha \subset \beta$. (Furthermore, β will not act at all unless $\alpha^\frown\langle 0 \rangle \subseteq \beta$, i.e., unless β guesses that $\Phi_e^A = U_e$.)

Each $\mathcal{P}_e$-strategy ξ is assigned to $D_\xi = \omega^{[2e]}$ for its thickness strategy. Each strategy ξ of type $\mathcal{R}$ or $\mathcal{S}$ is effectively assigned to an infinite recursive subset D_ξ of ω so that

(13) $$\bigsqcup_{\xi \text{ of type } \mathcal{R} \text{ or } \mathcal{S}} D_\xi = \bigcup_{e \in \omega} \omega^{[2e+1]}.$$

All $\mathcal{N}$-strategies $\gamma \supseteq \alpha^\frown\langle 0 \rangle$ (where α is a fixed $\mathcal{R}_e$-strategy) also help each $\mathcal{S}_{e,i}$-strategy β with $\alpha^\frown\langle 0 \rangle \subseteq \beta \subset \gamma$ build its part of the set B_e, so each γ is effectively assigned an infinite recursive subset E_α^γ such that for fixed α,

(14) $$\bigsqcup_{\substack{\gamma \supseteq \alpha^\frown\langle 0 \rangle \\ \gamma \text{ of type } \mathcal{N}}} E_\alpha^\gamma = \omega.$$

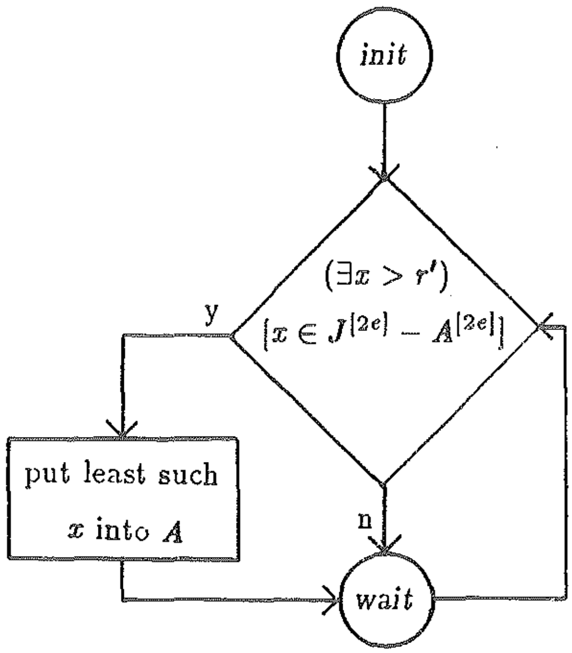

Diagram 2. The $\mathcal{P}$-strategy

Let also $r(\gamma)$ (or r, for short) denote the *A-restraint imposed by the $\mathcal{N}$-strategy* γ (as defined below), and

(15) $$r'(\xi) = \max\{\, r(\gamma) \mid \gamma < \xi \,\}$$

(or r', for short) the *A-restraint imposed on* ξ by all stronger strategies. (Recall that only $\mathcal{N}$-strategies impose restraint, so $r(\xi) = -1$ for all other strategies ξ.)

At each stage s, we will build substage by substage the approximation $\delta_s = \max\{\, \xi \mid \xi \text{ acts at stage } s \,\}$ to the true path $f \in [T]$ (where $|\delta_s| \le s$). We say s is a *ξ-stage* ($s \in S^\xi$) iff $\xi \subseteq \delta_s$. In the construction below, each strategy that acts at substage t of stage s will decide which strategy will act at substage $t+1$ (or whether we should go on to stage $s+1$, e.g., when $t = s$). $\emptyset$ will always be the strategy to act at substage 0. (When an $\mathcal{R}$- or an $\mathcal{S}$-strategy ξ lets an $\mathcal{N}$-strategy γ below it act first, then the action of γ will not count towards the definition of δ_s or as a separate substage.) Any strategy $\xi >_L \delta_s$ will be initialized as soon as δ_s has been defined far enough (i.e., at the least substage t at which $\delta_t[s] <_L \xi$).

The $\mathcal{P}$-strategies are the easiest to describe. They ensure that A is high. Recall that the r.e. set J codes $\emptyset''$ in the limit on the even rows. Then a $\mathcal{P}_e$-strategy ς acts as described in Diagram 2.

The strategy to play next will be $\varsigma^\frown\langle 0\rangle$ if $A_s^{[2e]} \ne A_t^{[2e]}$ where $t = \max\{\, t' < s \mid t' \in S^\varsigma \,\}$, and $\varsigma^\frown\langle 1\rangle$ otherwise.

Each $\mathcal{R}_e$-strategy α is responsible for building its version of the functional Θ_e, and it is the node where the construction of its version of the r.e. set B_e originates on the tree. Then α proceeds as described in Diagram 3.

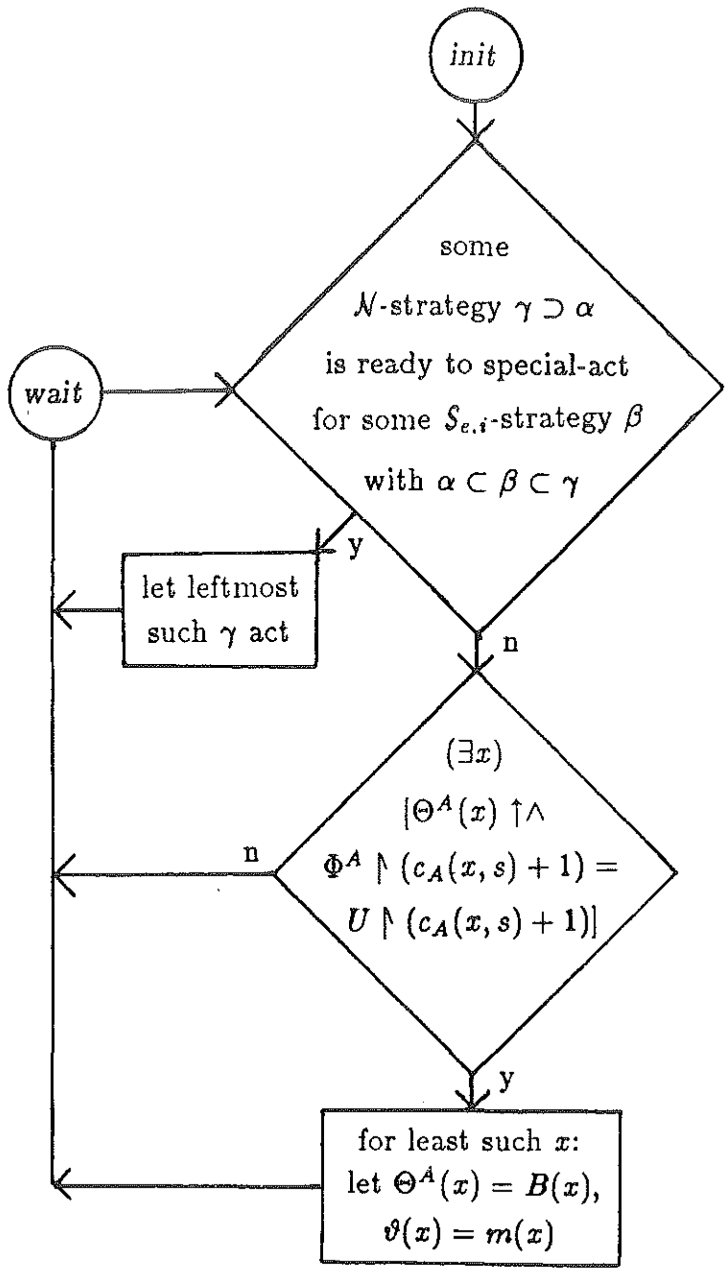

Diagram 3. The $\mathcal{R}$-strategy

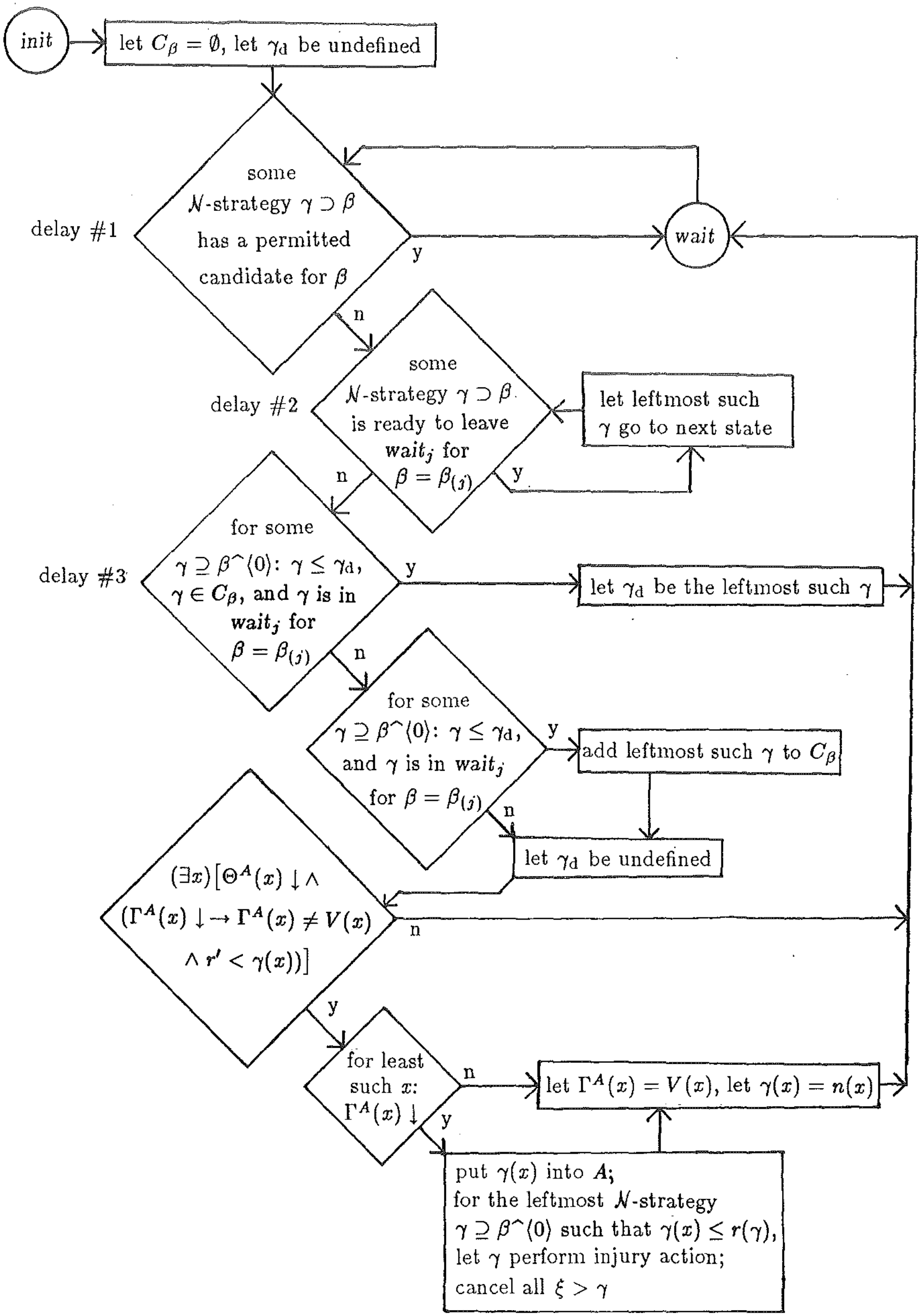

Diagram 4. The $\mathcal{S}$-strategy

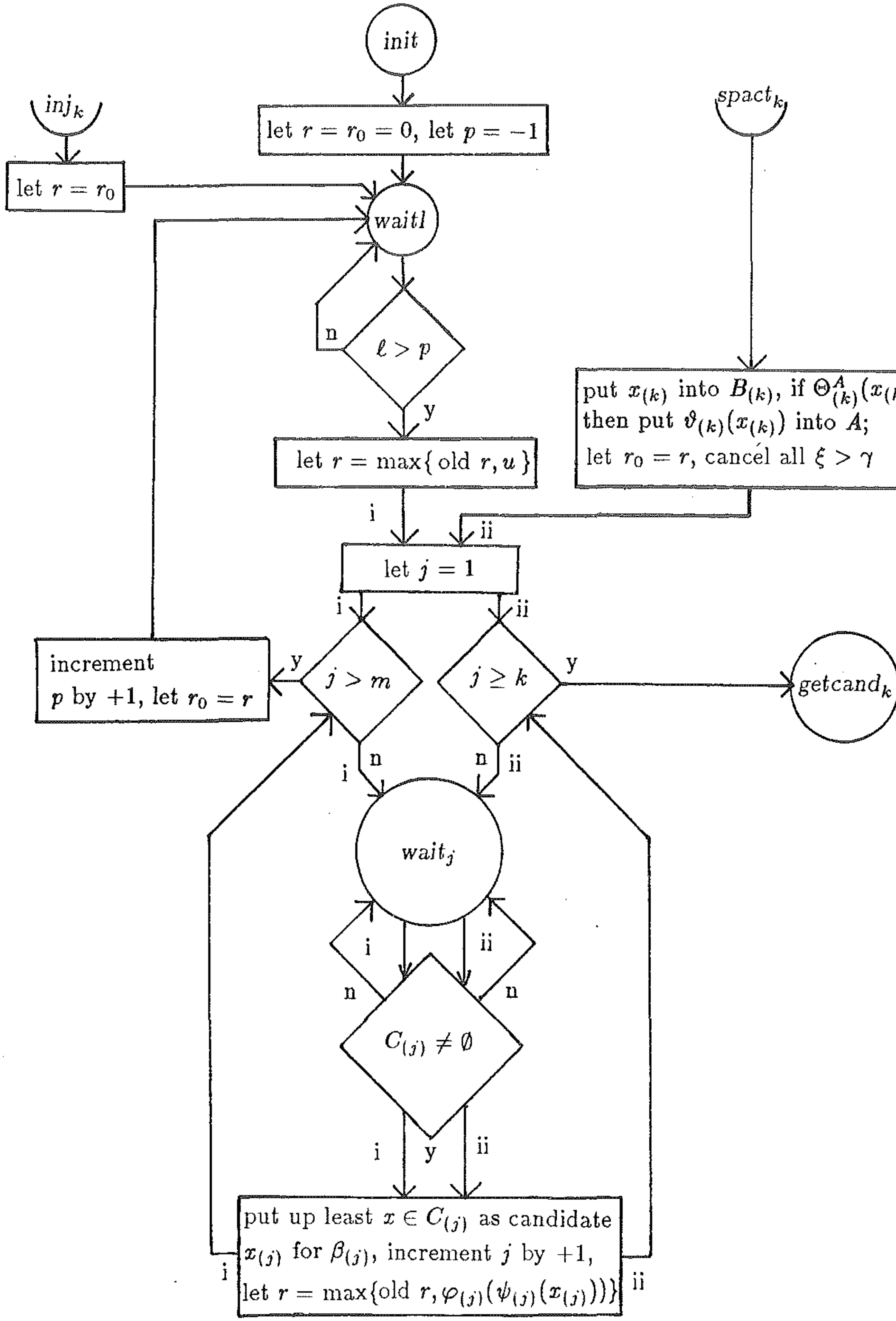

Diagram 5. The $\mathcal{N}$-strategy

Here $m(x)$ (the assigned use for $\Theta^A(x)$) is the least $y \in D_\alpha - A$ such that $y \geq$ previous values of $\vartheta(x)$ and greater than $\vartheta(x-1)$, $\varphi(c_A(x,s))$, and r'.

An $\mathcal{N}$-strategy $\gamma \supseteq \alpha^\frown\langle 0 \rangle$ is ready to *special-act* if:

(i) γ has put up a candidate $x_{(k)}$ for an $\mathcal{S}_{e,i}$-strategy $\beta_{(k)} \supseteq \alpha^\frown\langle 0 \rangle$ at a previous stage s_0;

(ii) γ has not been initialized since stage s_0;

(iii) no element entered $A \upharpoonright (r_{s_0}(\gamma)+1)$ since stage s_0, but $V_{(k)} \upharpoonright x_{(k)}$ has changed since stage s_0; and

(iv) no candidate for any $\beta_{(j)}$ with $j \leq k$ has been permitted since γ was initialized for the last time.

In this case, γ goes to $spact_k$ and on to the next state and gets a *permitted candidate* $x_{(k)}$ for $\beta_{(k)}$ through its special action (until γ is initialized if ever).

The strategy control will end the current stage if α lets some $\mathcal{N}$-strategy special-act. Otherwise, the next strategy to act will be $\alpha^\frown\langle 0 \rangle$ if α just (re)defined $\Theta^A(x)$ for some x, else it will be $\alpha^\frown\langle 1 \rangle$.

An $\mathcal{S}_{e,i}$-strategy β will *only* ever act if $\alpha^\frown\langle 0 \rangle \subseteq \beta$ for the $\mathcal{R}_e$-strategy $\alpha \subset \beta$. In this case, it will try to code V_e into A by building its version of $\Gamma_{e,i}$ to show $\Gamma_{e,i}^A = V_e$ unless some $\mathcal{N}$-strategy below it helps it to satisfy $\mathcal{S}_{e,i}$ in some other way. Therefore, β can be delayed in its action in various ways by $\mathcal{N}$-strategies below. An $\mathcal{S}_{e,i}$-strategy will thus act as described in Diagram 4.

Here $n(x)$ is the least $y \in D_\beta - A$ such that $y \geq$ previous values of $\gamma(x)$ and greater than $\gamma(x-1)$, $\vartheta(x)$, and r'.

An $\mathcal{N}$-strategy $\gamma \supseteq \beta^\frown\langle 0 \rangle$ performs *injury action* by going to inj_k (where $\beta = \beta_{(k)}$) and on to the next state.

Roughly speaking, γ_d is the strategy that caused delay #3 the last time β could act. (We agree that $\gamma \leq \gamma_d$ is satisfied vacuously if γ_d is undefined.) Its role is to eventually stop β if some $\mathcal{N}$-strategy below cannot find a candidate for β. Before γ can delay β, however, it has to be injured at least once (by the definition of C_β). We need C_β in Lemma 2 since for any s, $C_\beta[s]$ is finite and thus well-ordered, whereas $\bigcup_{s\in\omega} C_\beta[s]$ may not be well-ordered.

The next strategy to act will be $\beta^\frown\langle 0 \rangle$ if β (re)defined $\Gamma^A(x)$ for some x, else it will be $\beta^\frown\langle 1 \rangle$.

(It is worthwhile to intuitively distinguish the different delays for β here: Delay #1 is immediate and permanent and corresponds to the fact that $B \neq \Psi^U$. Delay #2 is always temporary, the $\mathcal{N}$-strategy below changes states, and then β resumes its action. Delay #3 is permanent again, but will only be activated eventually, corresponding to the outcome that $A \leq_T U$. If β is on the true path f and makes its Γ total, then each $\mathcal{N}$-strategy γ with $\beta \subset \gamma \subset f$ will eventually no longer be injured by β since γ's candidate protects γ against β.)

Finally, we will describe the most complicated of all strategies, the $\mathcal{N}$-strategy. Recall that an $\mathcal{N}$-strategy γ is trying to restrain A in order to ensure $\{e\}^A \neq K$. Towards the strategies $\xi > \gamma$, γ will use the usual Sacks preservation strategy; γ will have a guess about the $\mathcal{P}$-strategies $\varsigma \subset \gamma$; against the (potentially infinite) injury by the $\mathcal{S}_{e',i'}$-strategies β with $\beta^\frown\langle 0 \rangle \subseteq \gamma$, γ will try to put up candidates to show $\Psi_{i'}^{U_{e'}} \neq B_{e'}$. The strategy γ will thus proceed as described in Diagram 5.

Here, p, r, and r_0 are parameters defined in the diagram, roughly denoting the *protected length of agreement* of $K = \{e\}^A$, the *A-restraint imposed by* γ, and the *part of the A-restraint to preserve the protected length of agreement*, respectively.

The other parameters are defined as follows: We call a computation $\{e\}^A(x)\downarrow$ *γ-correct* iff

$$(16)\quad (\forall e' < \frac{|\gamma|}{4})(\forall z \in \omega^{[2e']} = D_{\gamma\restriction(4e'+1)}) \\ [\gamma(4e'+1) = 0 \wedge r'(\gamma \restriction (4e'+1)) < z < u(A;e,x) \rightarrow z \in A],$$

i.e., if all $\mathcal{P}$-strategies $\varsigma \subset \gamma$ that act infinitely often will not destroy the computation $\{e\}^A(x)\downarrow$. Then the *length of agreement* of $K = \{e\}^A$ is defined by

$$(17)\quad \ell = \max\{\, y \mid (\forall z < y)[K(z) = \{e\}^A(z)] \wedge \{e\}^A(y)\downarrow \\ \text{via } \gamma\text{-correct computations}\,\}.$$

The *use of the protected length of agreement* is

$$(18)\qquad u = \max\{\, u(A;e,y) \mid y \leq p+1 \,\}.$$

For the sake of simplicity, for fixed γ, we denote all $\mathcal{S}$-strategies such that $\beta_{(1)}{}^\frown\langle 0 \rangle \subset \beta_{(2)}{}^\frown\langle 0 \rangle \subset \cdots \subset \beta_{(m)}{}^\frown\langle 0 \rangle \subseteq \gamma$ by $\beta_{(1)}, \ldots, \beta_{(m)}$ (these are the strategies against which γ must put up a candidate), and all of the parameters of $\beta_{(j)}$ are temporarily denoted by $B_{(j)}$, $\Phi_{(j)}$ etc.

Let $\alpha_{(j)}$ be the $\mathcal{R}_{e_{(j)}}$-strategy such that $\alpha_{(j)} \subset \beta_{(j)}$. The *set $C_{(j)}$ of possible candidates for $\beta_{(j)}$* is defined as the set of all $y \in E^{\gamma}_{\alpha_{(j)}}$ such that:

(i) $y > r'$ and $y >$ any previous candidate that γ put up for $\beta_{(j)}$;
(ii) $\Psi_{(j)}^{U_{(j)}}(y) \downarrow = 0$;
(iii) $U_{(j)} \restriction (\psi_{(j)}(y)+1) = \Phi^A_{(j)} \restriction (\psi_{(j)}(y)+1)\downarrow$ via a γ-correct computation;
(iv) $\Theta^A_{(j)} \restriction (y+1)\downarrow$ via a γ-correct computation and $\vartheta_{(j)}(y) > r', r$; and
(v) $c_A(y,s) > \psi_{(j)}(y)$.

If γ changed states then all strategies $\xi > \gamma$ will be initialized, and a new stage is started. Otherwise, the next strategy to act will be $\gamma^\frown\langle \max\{r, r'\} \rangle$. (Recall that special action or injury action does not count as γ's turn, and that after special action the current stage is ended.)

(Intuitively, an $\mathcal{N}_e$-strategy tries to protect one by one the length of agreement of $K = \{e\}^A$ against stronger $\mathcal{S}$-strategies. Once it is in state *getcand$_k$* and thus has a permitted candidate for one of them, it assumes that it is to the left of the true path and will no longer protect longer lengths of agreement.)

5. The Verification. Let δ_s be the string of strategies that act at stage s (except for special action and injury action by the $\mathcal{N}$-strategies). Let $f = \liminf_s \delta_s$ be the true path on the tree T.

The verification consists of several lemmas:

LEMMA 1 (INJURY LEMMA). *No strategy ξ injures a strategy $\xi' < \xi$ by putting into A an element $x \leq r(\xi')$.*

PROOF: An $\mathcal{R}$-strategy does not put elements into A at all. The $\mathcal{P}$- and $\mathcal{S}$-strategies observe restraints by stronger strategies explicitly. Moreover, when an $\mathcal{N}$-strategy puts up a candidate, it is greater than stronger restraint so we only have to show that this restraint will not increase until the candidate is cancelled or put into A. But only the $\mathcal{N}$-strategies $\xi' < \xi$ impose stronger restraint. Whenever this restraint increases, some $\mathcal{N}$-strategy $\xi' < \xi$ has changed states, and therefore ξ must have been initialized. ∎

LEMMA 2 ($\mathcal{N}$-STRATEGY LEMMA). *Each $\mathcal{N}_e$-strategy $\gamma \subset f$ is injured at most finitely often, is eventually in state waitl (waiting for ℓ to increase), and $\lim_s \ell < \infty$ exists. (Thus $\lim_s r < \infty$ exists, $K \neq \{e\}^A$, and $\mathcal{N}_e$ is satisfied.)*

PROOF: First notice that any strategy $\xi <_L f$ acts only finitely often. This is trivial except for $\mathcal{N}$-strategies. But whenever an $\mathcal{N}$-strategy $\gamma <_L f$ performs special action or injury action, it will need $\gamma \subseteq \delta_s$ to act the next time.

We now use induction on $|\gamma|$ and the fact that $\gamma \leq \liminf_s \delta_s$. Let s_0 be minimal such that, after stage s_0, if any $\xi < \gamma$ acts then ξ is not an $\mathcal{N}$-strategy and $\xi \subset \gamma$, and such that every $\mathcal{N}$-strategy $\gamma' \subset \gamma$ is in state *waitl* and is not injured after stage s_0.

Thus, γ is initialized after stage s_0 only if some $\mathcal{S}$-strategy $\beta_{(j)}$ (as defined for γ) with $\beta_{(j)}{}^\frown\langle 0 \rangle \subseteq \gamma$ lets γ perform injury action. Since no $\mathcal{N}$-strategies $\gamma' <_L \gamma$ ever act after stage s_0, none of these will *start* delaying any $\mathcal{S}$-strategies $\beta_{(j)}$ (as defined for γ) after stage s_0 more than once (i.e., after they entered $C_{\beta_{(j)}}$); but $\beta_{(j)}{}^\frown\langle 0 \rangle \subset f$, and therefore eventually, say, after stage $s_1 \geq s_0$, none of these will ever delay any $\mathcal{S}$-strategy $\beta_{(j)}$. So after stage s_1, for all $j = 1, 2, \ldots, m$, we have that $\gamma_{\mathrm{d}}^{(j)} \geq \gamma$. Thus after stage s_1, once $\gamma \in C_{\beta_{(j)}}$, γ can delay $\beta_{(j)}$ until it has a candidate against it. γ will therefore eventually no longer be injured. (Recall that γ knows which elements will be put into A by $\mathcal{P}$-strategies $\zeta \subset \gamma$ after stage s_0.) But then as in the usual Sacks preservation strategy, $K = \{e\}^A$ would imply that K is recursive, so $\lim_s \ell < \infty$ exists and γ will eventually stop acting and be in state *waitl* forever (waiting for ℓ to increase). So $\lim_s r < \infty$ exists, $K \neq \{e\}^A$, and $\mathcal{N}_e$ is satisfied. ∎

LEMMA 3 ($\mathcal{P}$-STRATEGY LEMMA). *For all e, $A^{[2e]} =^* J^{[2e]}$. Thus A is high.*

PROOF: Only the $\mathcal{N}$-strategies impose restraint on A. Lemma 2 shows that this restraint is finite along the true path. ∎

LEMMA 4 (CORRECT Θ_e LEMMA). *If $U_e = \Phi_e^A$, then the $\mathcal{R}_e$-strategy $\alpha \subset f$ makes Θ_e total and $B_e = \Theta_e^A$.*

PROOF: Suppose by induction that after stage s_0, $\Theta_e^A \upharpoonright x$ has been defined A-correctly; that if strategy $\xi < \alpha$ acts then $\xi \subset \alpha$ and ξ is not an $\mathcal{N}$-strategy; that x is already a candidate for the $\mathcal{N}$-strategy $\gamma \supset \alpha$ (if it ever will be) where $x \in E_\alpha^\gamma$; and that $\Phi^A \upharpoonright (c_A(x)+1)$ has settled down. But then $m(x)$ changes at most once, namely, when γ puts $\vartheta(x)$ into A, and afterwards x will never again be a candidate. So $m(x)$ will eventually be constant, and thus $\Theta^A(x)$ will eventually be defined A-correctly. Thus Θ^A is total. Furthermore, $B = \Theta^A$ since B only changes on x when $\Theta^A(x)$ is or becomes undefined. ∎

LEMMA 5 (DELAY #3 LEMMA). *For any $\mathcal{S}$-strategy $\beta \subset f$, if β is delayed by delay #3 cofinitely often, then eventually β is always delayed by delay #3 by some fixed $\mathcal{N}$-strategy $\gamma = \lim_s \gamma_{\rm d}$.*

PROOF: Suppose β is not initialized after stage s_0. If β is delayed cofinitely often by delay #3, then $C_{\beta,\infty} = \bigcup_{s\in\omega} C_\beta[s]$ is finite and thus well-ordered. Let γ_0 be the leftmost $\gamma \in C_{\beta,\infty}$ that causes delay #3 for β infinitely often. Then $\gamma_0 = \lim_s \gamma_{\rm d}[s]$ since whenever $\gamma_{\rm d}[s] > \gamma_0$ and later $\gamma_{\rm d}[s'] = \gamma_0$ then β is not delayed by delay #3 at least once between stages s and s' by the arrangement of delay #3. (This is the reason for having $\gamma_{\rm d}$ and C_β in this construction.) ∎

LEMMA 6 (CORRECT Γ LEMMA). *If $U_e = \Phi_e^A$ then the $S_{e,i}$-strategy $\beta \subset f$ makes its version of $\Gamma_{e,i}^A$ total and $V = \Gamma_{e,i}$ unless β is eventually permanently delayed by one fixed $\mathcal{N}$-strategy $\gamma \supseteq \beta^\frown\langle 0\rangle$ through delay #1 or delay #3.*

PROOF: Suppose that if any strategy $\xi < \beta$ acts after stage s_0 then $\xi \subset \beta$ and ξ is not an $\mathcal{N}$-strategy; and that no $\xi \leq \beta$ is initialized after stage s_0. Then β is never initialized after stage s_0, and so either it is permanently delayed by one fixed $\mathcal{N}$-strategy (by Lemma 5 for delay #3 and by the construction for delay #1) and $\beta^\frown\langle 1\rangle \subset f$; or β is not delayed at infinitely many β-stages. (Recall that delay #2 was only temporary.) In the latter case, β can define or redefine $\Gamma_{e,i}$ infinitely often.

Suppose by induction that after stage $s_1 \geq s_0$, $\Gamma_{e,i}^A \restriction x$ has been defined A-correctly; and that $V_e \restriction (x+1) = V_{e,s} \restriction (x+1)$ and $\Theta_e^A \restriction (x+1) \downarrow$ A-correctly. Then $n(x)$ is constant after stage s_1, so $\Gamma_{e,i}^A(x)$ will eventually be defined A-correctly. Thus $\Gamma_{e,i}$ is total. Furthermore, $V_e(x) = \Gamma_{e,i}^A(x)$ at least for all $x > \lim_s r'[s]$ (since $\gamma(x) \geq x$). ∎

LEMMA 7 (CORRECT $\Xi_e/\mathcal{R}$-STRATEGY LEMMA). *Let $\alpha \subset f$ be the $\mathcal{R}_e$-strategy. Then the version of B_e that originates at α is recursive in V_e by direct permitting on α-stages. Thus $\mathcal{R}_e$ is satisfied through α's versions of Θ_e, Ξ_e and B_e (by this lemma and Lemma 4).*

PROOF: Element x can enter B_e only as a candidate through special action of the $\mathcal{N}$-strategies $\gamma \supset \alpha$. This special action can only occur until the first α-stage s at which $V \restriction x = V[s] \restriction x$. ∎

LEMMA 8 ($\mathcal{S}$-STRATEGY LEMMA). *Let $\alpha \subset f$ be the $\mathcal{R}_e$-strategy. If, for α's version of B_e and fixed i, $U_e = \Phi_e^A$, $U_e <_T A$, and $B_e = \Psi_i^{U_e}$ then the $S_{e,i}$-strategy $\beta \subset f$ is not eventually permanently delayed by $\mathcal{N}$-strategies. (Thus, by Lemma 6, $\Gamma_{e,i}$ is total and $V_e = \Gamma_{e,i}^A$, so $S_{e,i}$ is satisfied.)*

PROOF: By Lemma 5, we only have to show that no single $\mathcal{N}$-strategy γ delays β forever. This can only happen if $\gamma \supseteq \beta^\frown\langle 0\rangle$ and $\beta^\frown\langle 1\rangle \subset f$. Suppose that after stage s_0, $\delta_s \geq \beta^\frown\langle 1\rangle$, that no strategy injures γ ever again (since otherwise γ cannot delay β at the next β-stage), and that γ does not act ever again.

If γ delays β by delay #1 then $B_e \neq \Psi_i^{U_e}$ through the permitted candidate since γ is no longer injured. If γ delays β by delay #3 then we show that $A \leq_T U_e$ as follows (this defines $\Delta_{e,i}$ implicitly): γ delays β because it cannot find a candidate for it. Let $\tilde{C}$ be the set of all $y \in E_\alpha^\gamma - B_e$ (where α is the $\mathcal{R}_e$-strategy $\alpha \subset \beta$) such that (at some stage $s > s_0$):

(i) $y > r'(\gamma)[s]$ and $y >$ any previous candidate that γ put up for $\beta_{(j)}$;

(ii) $\Psi^U(y)[s] \downarrow = 0$;

(iii) $U \upharpoonright (\psi(y)+1)[s] = \Phi^A \upharpoonright (\psi(y)+1)[s] \downarrow$ via a γ-correct computation; and

(iv) $\Theta^A[s] \upharpoonright (y+1) \downarrow$ via a γ-correct computation and $\vartheta[s](y) > r'(\gamma)[s], r(\gamma)[s]$.

Since $U = \Phi^A$ and $B = \Psi^U$ and r and r' settle down, this is an infinite U-recursive set, but then $C = \tilde{C} \cap \{ y \mid c_A(y) > \psi(y) \}$ has to be finite, or else the $\mathcal{N}$-strategy γ would find a candidate eventually. Since ψ is total, we have that $\psi \leq_T U$, and ψ dominates c_A on the set $\tilde{C}$. Therefore, A is recursive in U. ∎

This concludes the proof of the theorem. ∎

References

[AS84] K. Ambos-Spies, *On Pairs of Recursively Enumerable Degrees*, Trans. Amer. Math. Soc. **283** (1984), 507–531.

[Cota] S.B. Cooper, *A Jump Class of Non-Cappable Degrees*, J. Symbolic Logic (to appear).

[La66] A.H. Lachlan, *Lower Bounds for Pairs of Recursively Enumerable Degrees*, Proc. London Math. Soc. **16** (1966), 537–569.

[Sa63] G.E. Sacks, *On the Degrees Less than* $\mathbf{0}'$, Annals of Mathematics **77** (1963), 211–231.

[Sa64] ———, *The Recursively Enumerable Degrees Are Dense*, Annals of Mathematics **80** (1964), 300–312.

[Sh65] J.R. Shoenfield, *Application of Model Theory to Degrees of Unsolvability*, in "Symposium on the Theory of Models", Ed. J.W. Addison, L. Henkin, and A. Tarski, North Holland, Amsterdam, pp. 359–363.

[Shta] R.A. Shore, *A Non-Inversion Theorem for the Jump Operator*, Annals of Pure and Applied Logic (to appear).

[So80] R.I. Soare, *Fundamental Methods for Constructing Recursively Enumerable Degrees*, in "Recursion Theory: Its Generalizations and Applications", Proceedings of Logic Colloquium '79, Leeds, August 1979, Ed. F.R. Drake and S.S. Wainer, Cambridge University Press, Cambridge, 1980.

[So85] ———, *Tree Arguments in Recursion Theory and the* $\mathbf{0}'''$*-Priority Method in Recursion Theory*, in "Proceedings of Symposia in Pure Mathematics No. 42", Amer. Math. Soc., Providence, pp. 53–106.

[Sota] ———, "Recursively Enumerable Sets and Degrees", Springer-Verlag, Heidelberg (to appear).

[Ya66] C.E.M. Yates, *A Minimal Pair of Recursively Enumerable Degrees*, J. Symbolic Logic **31** (1966), 159–168.

Yale University, New Haven, CT 06520