A Limit on Relative Genericity in the Recursively Enumerable Sets

STEFFEN LEMPP

THEODORE A. SLAMAN

ABSTRACT. Work in the setting of the recursively enumerable sets and their Turing degrees. A set X is *low* if X', its Turing jump, is recursive in $\emptyset'$ and *high* if X' computes $\emptyset''$. Attempting to find a property between being low and being recursive, Bickford and Mills produced the following definition. W is *deep*, if for each recursively enumerable set A, the jump of $A \oplus W$ is recursive in the jump of A. We prove that there are no deep degrees other than the recursive one.

Given a set W, we enumerate a set A and approximate its jump. The construction of A is governed by strategies, indexed by the Turing functionals Φ. Simplifying the situation, a typical strategy converts a failure to recursively compute W into a constraint on the enumeration of A, so that $(W \oplus A)'$ is forced to disagree with $\Phi(-; A')$. The conversion has some ambiguity; in particular, A cannot be found uniformly from W.

We also show that there is a "moderately" deep degree: There is a low non-zero degree whose join with any other low degree is not high.[1]

§1. INTRODUCTION

There is a strong similarity between building a generic real (subset of the integers), by meeting dense subsets of a partially ordered set, and

The results appearing here form a section of Lempp's Ph.D. thesis, The University of Chicago, 1986. The authors would like to thank R. I. Soare, who is also Lempp's thesis advisor, for encouraging this work. Slaman was supported by N.S.F. grant DMS-8601856 and Presidential Young Investigator Award DMS-8451748.

[1] Harrington has recently shown that there is a low non-zero degree whose join with any other low degree is also low.

recursively enumerating a real, by simultaneously executing a family of recursive strategies. The analogy is exact provided that the forcing partial ordering is recursive and genericity is only required relative to a uniformly recursive family of dense sets. Otherwise, priority methods diverge from forcing techniques. To take an obvious example, it is not possible to recursively enumerate a set of integers so that it is different from every Σ^0_1-set, even though the diagonalizing conditions are dense. Instead of meeting every dense set, as does a fully generic real, the real enumerated during a priority construction meets sets that are dense in its enumeration. In fact, this is the typical effect of a strategy on the construction of a recursively enumerable set.

Cohen forcing in the context of recursively enumerable sets has been systematically studied and well understood by Maass [**Ms82**], Jockusch [**Jo85**], and others. In their work, reals were enumerated so as to meet certain dense subsets of the Cohen partial order. The reals produced were then shown to have many of the recursion theoretic properties associated with Cohen generic reals. To cite an example, let G be Maass generic (roughly speaking, its enumeration is generic with respect to all primitive recursive sets), and let G' be its Turing jump. A $\Sigma^0_1(G)$-statement is true if and only if it is forced by some strategy; further, the forcing relation for $\Sigma^0_1(G)$-statements is Σ^0_1. A $\Pi^0_1(G)$-statement is true if and only if its associated strategy cannot force its negation. Given that G is produced by a recursive construction that uses only strategies with outcomes recursive in $\emptyset'$, these two facts show that G' is recursive in $\emptyset'$.

In this paper, we will examine a Σ^0_2-aspect of Cohen genericity. Namely, how close can a recursively enumerable set come to being Cohen generic relative to every recursively enumerable set? For other results addressing the same issue, see Shore [Shta] or Soare-Stob [SS82].

Given two sets of integers A and B, let $A \oplus B$ denote their effective disjoint union. Bickford and Mills defined a recursively enumerable set W to be *deep* if for every recursively enumerable set A, the jump of $A \oplus W$ is recursive in A'. Following the guide of Cohen genericity, we might attempt to enumerate a set W, make W sufficiently Cohen generic to be different from every recursive set and ensure W's being deep by using strategies to decide $\Sigma^0_1(A \oplus W)$-statements in a way that is recursive in A'. For a single formula, it is not hard to find a strategy that satisfies the third condition and is compatible with W's not being recursive; in fact, the strategies fit together to show that, for every recursively enumerable set A, there is a recursively enumerable set W such that W is not recursive in A (if A is incomplete) and $(A \oplus W)'$ is recursive in A'.

The obstacle appears in the attempt to make W simultaneously deep with regard to two sets A and $\hat{A}$. The way that a strategy decides a $\Sigma^0_1(A \oplus W)$-statement is recursive in A' but not necessarily recursive in $\hat{A}'$. Similarly, A' cannot compute those steps taken for the sake of genericity relative to $\hat{A}$. This obstacle is insurmountable; in section §3, we show that if W is recursively enumerable and not recursive then W is not deep. In the proof of the theorem, we find strategies showing that the obstacle mentioned above is completely general. That is, no recursively enumerable set is sufficiently Cohen generic for the Maass style calculation of the jump to apply relative to an arbitrary recursively enumerable set.

The proof is organized as follows. Suppose that W is a recursively enumerable set. First, we enumerate a set A and functional Γ. Suppose that

W meets enough dense sets to be not recursive and to satisfy

$$(1) \qquad \Gamma(-;(W \oplus A)') = \Psi(-;A').$$

for all Ψ. The key feature of the argument is that the dense sets associated with W's not being recursive involve numbers entering W. The dense sets associated with equation (1) involve W's enumeration's being timed so that numbers enter W only when other numbers enter A. We synchronize the enumeration of another pair $\hat{A}$ and so that numbers enter $\hat{A}$ only when numbers are not entering A. Thus, it is impossible for numbers to enter W only when $\hat{A}$ changes. The combination of W's not being recursive and also not being fully generic relative to A provides the vehicle by which $(W \oplus \hat{A})'$ coherently computes a set not recursive in $\hat{A}'$. Namely, $\hat{\Gamma}(-;(W \oplus \hat{A})')$ meets dense sets relative to $\hat{A}'$, making it diagonalize against functions recursive in $\hat{A}'$, during the stages when W meets those sets making W not recursive.

A set X is *low* if X' is recursive in $\emptyset'$; X is *high* if X' computes $\emptyset''$. Because of the possibility of infinitary outcomes in the strategies used above, we are unable to conclude that the sets A and $\hat{A}$ are low. In section §4, we show that there is a recursively enumerable nonrecursive set W such that its join with any low recursively enumerable set is not high. Recently, Harrington improved this result to show that there is a recursively enumerable nonrecursive set W whose join with any low recursively enumerable set is again low. In both constructions, the set being enumerated is only required to be deep relative to low sets. This makes it possible to meet dense sets that are not accessible relative to an arbitrary recursively enumerable set.

§2. Notation

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

We work in the context of sets and functions on ω, the set of natural numbers $\{0, 1, 2, 3, \dots\}$. 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. $f \upharpoonright x$ denotes f restricted to arguments less than x, likewise for sets.

We let $A \subset B$ denote that $A \subseteq B$ but $A \neq B$. $A \sqcup B$ 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. $A^{[k]} = \{y \mid \langle y, k \rangle \in A\}$ denotes the kth "row" of A; and $A^{[<l]} = \bigcup_{k<l} A^{[k]}$.

The logical connectives "and" and "or" will be denoted by & 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* of integers; $|\sigma|$ the *length* of σ; $\sigma^\frown\tau$ 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, $\sigma \upharpoonright i = \tau \upharpoonright i$ 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 $\{ p \in \Lambda^\omega \mid (\forall n)[p \upharpoonright n \in T] \}$.

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 a recursively enumerable 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]).

§3. Deep Degrees

Bickford and Mills defined the notion of a deep degree:

DEFINITION. A recursively enumerable degree $\mathbf{w}$ is *deep* if for all recursively enumerable degrees $\mathbf{a}$,

$$\mathbf{a}' = (\mathbf{a} \cup \mathbf{w})'. \tag{1}$$

They raised the question of whether a nonrecursive deep degree exists.

THEOREM. *The only deep recursively enumerable degree is the recursive degree.*

PROOF: For each recursively enumerable set W, we have to construct a recursively enumerable set A such that

$$\hat{\mathcal{R}} : W \leq_T \emptyset \vee A' <_T (A \oplus W)'. \tag{2}$$

Let us first show that A cannot be built uniformly in W. Suppose there is a recursive function f such that for all e,

$$W_e \leq_T \emptyset \vee W'_{f(e)} <_T (W_{f(e)} \oplus W_e)'. \tag{3}$$

We will show that there is a recursive function g such that for all e

$$\text{(4A)} \qquad (W_e \oplus W_{g(e)})' \equiv_T W_e',$$

$$\text{(4B)} \quad (W_e <_T \emptyset' \implies W_{g(e)} \not\leq_T W_e) \,\&\, (W_e \equiv_T \emptyset' \implies W_{g(e)} \equiv_T \emptyset').$$

Now pick a fixed point e_0 for gf by the Recursion Theorem. Then

$$\text{(5)} \qquad (W_{e_0} \oplus W_{f(e_0)})' \equiv_T (W_{gf(e_0)} \oplus W_{f(e_0)})' \equiv_T W_{f(e_0)}'.$$

By our assumption (3) on f (which was supposed to pick a counterexample to W_{e_0} deep), W_{e_0} has to be recursive. Therefore, $W_{gf(e_0)}$ is also recursive. This contradicts our claim (4) about g (which is supposed to build nonrecursive sets).

The proof of (4) is a simple infinite injury argument. For given $W = W_e$, we have to uniformly build $A = W_{g(e)}$. To satisfy $(W \oplus A)' \leq_T W'$, we use the Sacks preservation strategy (as in Sacks [**Sa63a**]); it preserves all possible computations to keep $(W \oplus A)'$ down; its restraint drops on W-true stages. In the attempt to satisfy $A \not\leq_T W$, we use the Sacks coding strategy, trying to code K into A (as in Sacks [**Sa64**]). Note that this strategy makes A complete if W is complete.

The Requirements and the Basic Module. Fix a recursively enumerable set W. We show (2) by building a functional Γ such that for all Ψ, $\lim_s \Gamma^{A\oplus W}(-,s)$ is not equal to $\lim_v \Psi^A(-,v)$. The construction of the counterexample to W's being deep is not uniform. We will build a pair (A, Γ) consisting of a recursively enumerable set A and a functional Γ, and a sequence $\{\,(\hat{A}_\Psi, \hat{\Gamma}_\Psi)\,\}_{\Psi \text{ functional}}$ of such pairs such that if A, Γ, and Ψ_0 do not satisfy the above inequality, then $(\hat{A}_{\Psi_0}, \hat{\Gamma}_{\Psi_0})$ will be the desired counterexample satisfying (2). The requirements will thus be as follows (for all pairs of functionals $(\Psi, \hat{\Psi})$):

$$\mathcal{R}_{\Psi,\hat{\Psi}} : W \leq_T \emptyset \vee \lim_v \Psi^A(-,v) \neq \lim_s \Gamma^{A\oplus W}(-,s)$$

$$\vee \lim_v \hat{\Psi}^{\hat{A}_\Psi}(-,v) \neq \lim_s \hat{\Gamma}_\Psi^{\hat{A}_\Psi \oplus W}(-,s). \tag{6}$$

Once we have shown that $W \leq_T \emptyset$ through one strategy, the requirements of lower priority need not be satisfied. (We will suppress the index Ψ on $\hat{A}_\Psi$ and $\hat{\Gamma}_\Psi$.)

The basic idea for the proof is to either observe changes in W often enough to make Γ (or $\hat{\Gamma}$) different from Ψ (or $\hat{\Psi}$) in the limit, or else to build an implicit recursive functional $\Delta_{\Psi,\hat{\Psi}}$ (or Δ, for short) to show that W is recursive via Δ.

The highest priority here is to make Γ (and all $\hat{\Gamma}$) total and to ensure that $\lim_s \Gamma^{A\oplus W}(-,s)$ (and all $\lim_s \hat{\Gamma}^{\hat{A}\oplus W}(-,s)$) exist.

To ensure the former for Γ, we will define $\Gamma^{A\oplus W}(x,s)$ at stage s. At stage 0, we set $\Gamma^{A\oplus W}(x,0) = 0$ and its use $\gamma(x,0) = 0$. If, at a stage $s' > s$, $\Gamma^{A\oplus W}(x,s)$ becomes undefined because of a change in A or W, we will redefine it by the end of stage s'. This will either be done explicitly by a strategy on the tree, or implicitly at the end of stage s', when the strategy control sets $\Gamma^{A\oplus W}(x,s) = \Gamma^{A\oplus W}(x,s-1)$ and $\gamma(x,s) = \gamma(x,s-1)$. We ensure that $\Gamma^{A\oplus W}(x,s)$ is redefined only finitely often by setting the use $\gamma(x,s)$ only equal to 0 and at most one other number.

To ensure that the limit of Γ exists, we commit ourselves that for all x and s, $\Gamma^{A\oplus W}(x,s) \leq \Gamma^{A\oplus W}(x,s+1) \leq 1$. So actually $\lim_s \Gamma^{A\oplus W}(-,s)$ will be $\Sigma_1^{A\oplus W}$. (There will be one modification later.) We do the same for the $\hat{\Gamma}_\Psi$.

For some fixed i, the basic module of a strategy (for one $\mathcal{R}_{\Psi,\hat{\Psi}}$) first tries to make $\lim_v \Psi^A(i,v)$ different from $\lim_s \Gamma^{A\oplus W}(i,s)$ by changing $\Gamma^{A\oplus W}(i,s)$ whenever $\Psi^A(i,v)$ catches up. If this fails, it tries to show that $\lim_v \Psi^A(i,v)$ does not exist by restraining A while resetting $\Gamma^{A\oplus W}(i,s) = 0$ for all s. The latter requires infinitely many numbers to go into W or A. If Γ has to be corrected cofinitely often by putting numbers into A (thus not enabling us to show that $\lim_v \Psi^A(i,v)$ does not exist) then we indirectly achieve temporary restraint on W. So the third possibility for the outcome is to combine this strategy with the same strategy for $\hat{A}$ to show that W is recursive.

The basic module can thus informally be described as follows (call this the A-side of the module):

(i) fix a candidate i (for $\lim_s \Gamma^{A\oplus W}(i,s) \neq \lim_v \Psi^A(i,v)$),

(ii) start setting $\Gamma^{A\oplus W}(i,s) = 0$ (until (iii) holds) at each stage s,

(iii) wait for $\Psi^A(i,v_0) = 0$ for some v_0 (at stage s_1, say),

(iv) impose A-restraint on $A \upharpoonright (\psi(i,v_0)+1)$,

(v) start setting $\Gamma^{A\oplus W}(i,s) = 1$ with $\gamma(i,s) > \psi(i,v_0)$ (until (ix) or (x) holds) at each stage s,

(vi) wait for $\Psi^A(i,v_1) = 1$ for some $v_1 > v_0$ (at stage s_2, say),

(vii) impose A-restraint on $A \upharpoonright (\psi(i,v_1)+1)$,

(Notice that we have now put a *squeeze* on our opponent: either $W \upharpoonright (\gamma(i,s_1)+1)$ changes, and we can reset $\Gamma^{A\oplus W}(i,s') = 0$ (for all $s' \geq s_1$) while $\Psi^A(i,-)$ has a ***flip*** (a switch from 0 to 1 back to 0), which we preserve; or else $W \upharpoonright (\gamma(i,s_1)+1)$ remains unchanged, which constitutes a step towards showing that W is recursive. In the second case, the effect is that we temporarily restrain W until we reset $\Gamma^{A\oplus W}(i,s') = 0$ (for all $s' \geq s$) by changing A below $\gamma(i,s_1)$.

The key idea is to run a copy of the module (ii)-(vii) (the $\hat{A}$-side) until this copy restrains W in a similar way. Our strategy threatens to compute W recursively by restraining it by the $\hat{A}$-side while $\Gamma^{A\oplus W}(i,s)=0$, and by the A-side while $\hat{\Gamma}^{\hat{A}\oplus W}(\hat{\imath},s)=0$.)

(viii) start the $\hat{A}$-side at (i) or restart the $\hat{A}$-side at (ii) (until (ix) or (x) holds),

(ix) if $W_{s_2} \upharpoonright (\gamma(i,s_1)+1) \neq W_s \upharpoonright (\gamma(i,s_1)+1)$ at stage s, then immediately reset $\Gamma^{A\oplus W}(i,s')=0$ for $s_1 \leq s' \leq s$, initialize the $\hat{A}$-side, and go to (ii) (looking for a new v_0 greater than the current v_1),

(x) if the $\hat{A}$-side reaches (vii), then stop it, put $\gamma(i,s_1)$ into A, reset $\Gamma^{A\oplus W}(i,s')=0$ for $s_1 \leq s' \leq s$, cancel the part of the A-restraint for preserving $A \upharpoonright (\psi(i,v_0)+1)$ and $A \upharpoonright (\psi(i,v_1)+1)$, and restart the A-side at (ii).

We will for this proof tacitly assume that for all x and v, $\psi(x,v) \leq \psi(x,v+1)$ (and likewise for $\hat{\psi}$).

Continuing in this informal way, let us verify that the basic module satisfies the requirement.

The outcomes can be classified as follows:

(a) finitary: One of the sides is waiting forever at (iii) or (vi) for $\Psi^A(i,-)$ (or $\hat{\Psi}^{\hat{A}}(\hat{\imath},-)$) to change. Then, if the limit for Ψ (or $\hat{\Psi}$) exists at all, it must be unequal to the limit of Γ (or $\hat{\Gamma}$).

(b) Ψ-flip: The A-side gets infinitely many W-changes at (ix). Then $\lim_v \Psi^A(i,v)$ cannot exist since we ensured infinitely many flips via A-restraint.

(c) $\hat{\Psi}$-flip: The $\hat{A}$-side gets infinitely many W-changes at (ix), the A-side only finitely many. Then $\lim_v \hat{\Psi}^{\hat{A}}(\hat{\imath},v)$ does not exist. Note that the candidate $\hat{\imath}$ settles down, and that $\lim_s \Gamma^{A\oplus W}(i,s)$ still exists since

$\Gamma^{A \oplus W}(i, s)$ is ultimately set to 0 for every s.

(d) recursive outcome: Both the A- and the $\hat{A}$-side get only finitely many W-changes at (ix) but both sides change states infinitely often. Then we will show that W is recursive. In this case, we will not need any other strategies to rule out W's being a nontrivial deep degree. The construction need not continue past this point, so we do not put this outcome on the tree.

The full module only requires two minor modifications:

1) If a strategy α has outcome (b) (or (c)) then the A-restraint (or $\hat{A}$-restraint, respectively) that α imposes on a weaker strategy β below this outcome on the tree tends to infinity. So β has to be able to injure α in some controlled way (*explicit injury feature*). But notice that α has some flexibility in preserving Ψ-flips (or $\hat{\Psi}$-flips). For each m, α can afford to have its mth flip injured finitely often before preserving it permanently. Then α will be able to preserve infinitely many flips, if it encounters infinitely many opportunities. β may have to put elements into A (or $\hat{A}$) in order to reset Γ (or $\hat{\Gamma}$), so β has to delay setting Γ (or $\hat{\Gamma}$) equal to 1 until it would be allowed to change α's flip and reset its own computation, if necessary (*delay feature*). When β assumes infinitely many Ψ-flips (or $\hat{\Psi}$-flips) for α, β can afford to wait.

2) Whenever a strategy α puts an element into A (or $\hat{A}$), a strategy β below it may be injured. However, the set that α puts in can be made strictly increasing, so β (if it is below outcome (b) or (c) of α) will wait until the part of A (or $\hat{A}$) it wants to work with is cleared of possible attention from α (*postponement feature*). β assumes that the numbers enumerated by α increase to infinity, so again β can afford to wait.

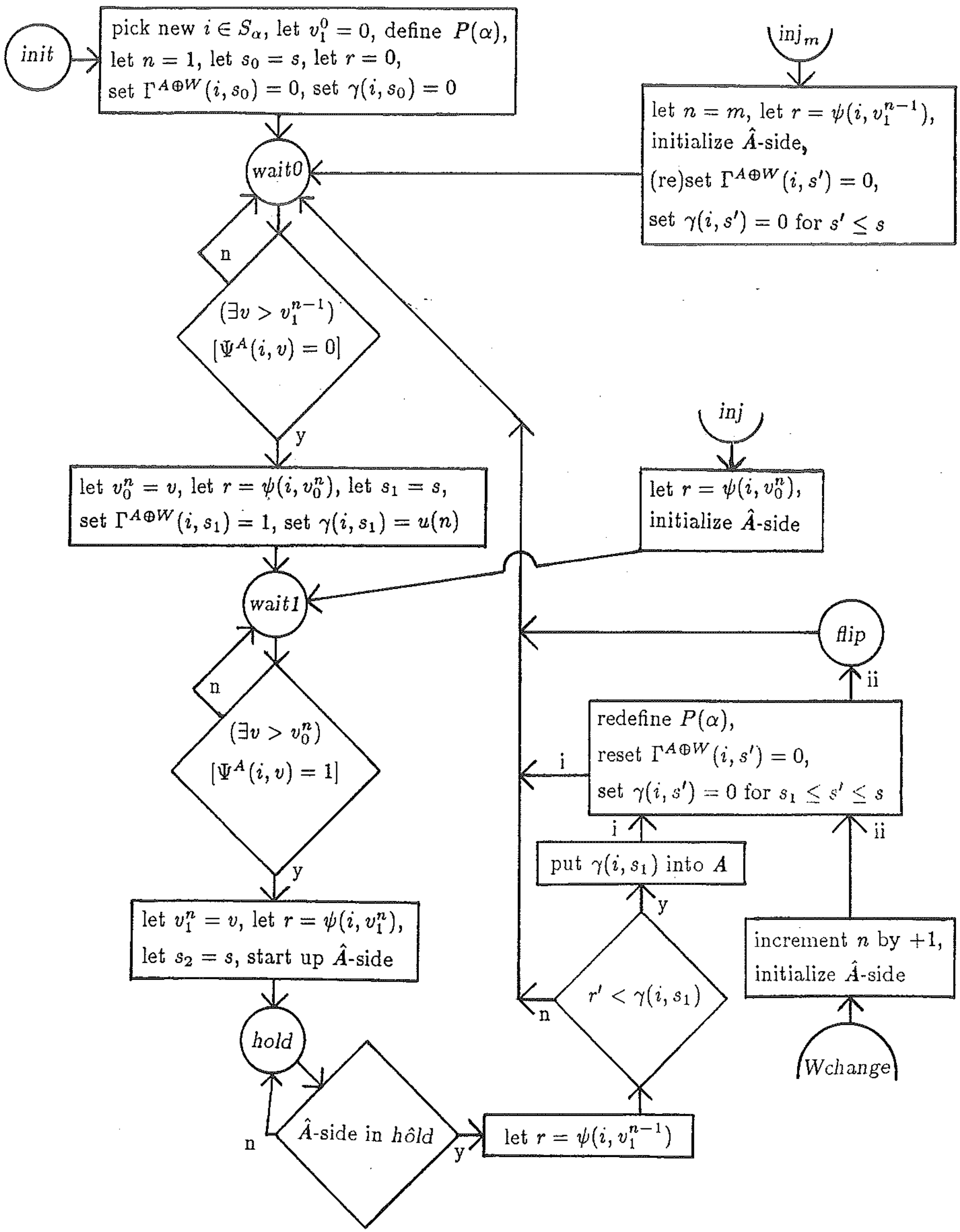

Diagram 1. The A-side

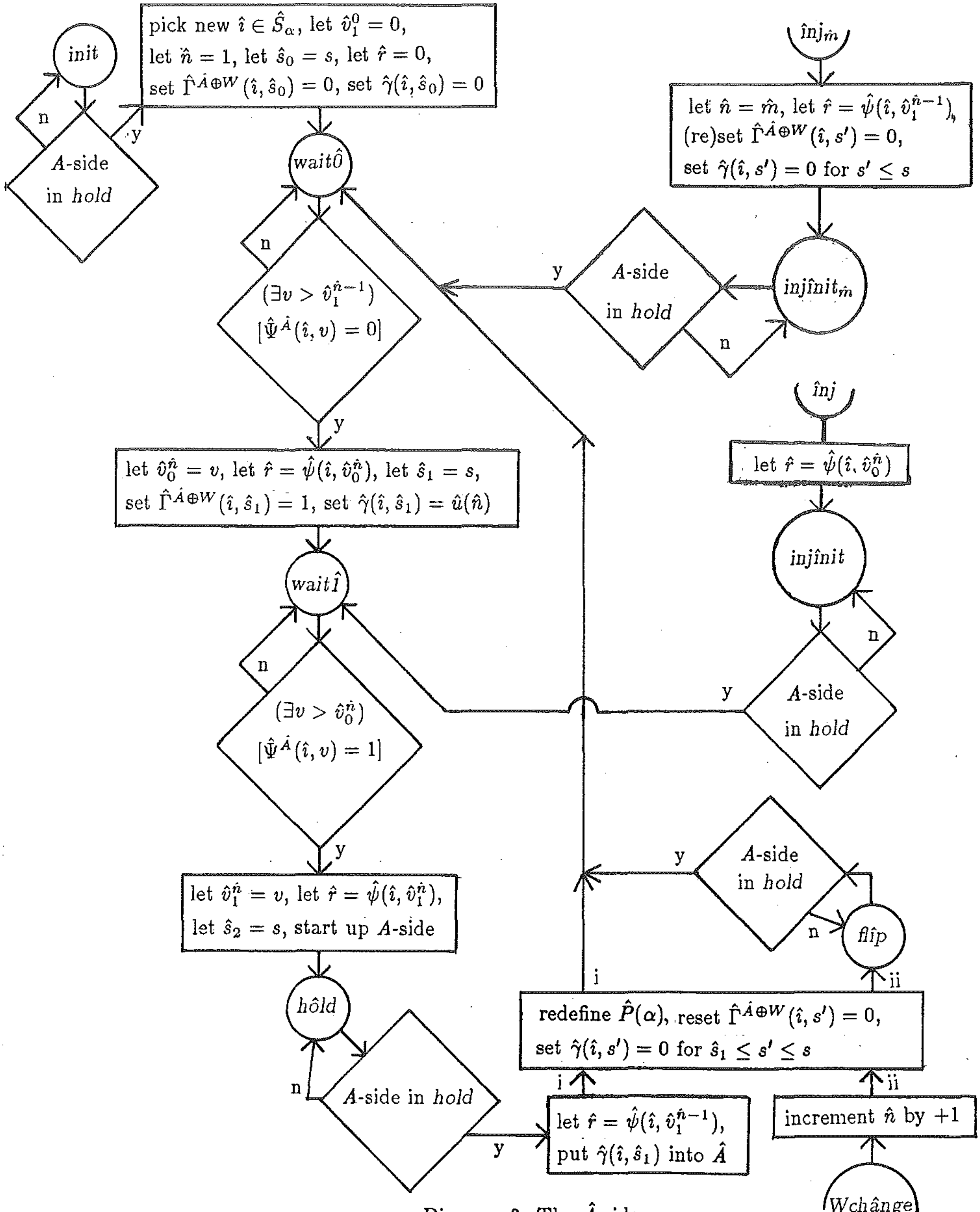

Diagram 2. The $\hat A$-side

The Full Construction. We will first describe the tree of strategies, then the full module for each strategy, and finally the strategy control, which supervises the interaction between the strategies.

Let $\Lambda = \{\, \mathit{flip} <_\Lambda \hat{\mathit{flip}} <_\Lambda \mathit{fin} \,\}$ be the set of outcomes. Notice that these outcomes correspond to the outcomes (b), (c), and (a), respectively, of the basic module above, that we collapsed all finitary outcomes into one, and that outcome (d) of the basic module will not be put on the tree since then this one strategy will satisfy the overall requirement (2). Now let $T = \Lambda^{<\omega}$ be the tree of strategies. Fix an effective 1–1 correspondence between all requirements $\mathcal{R}_{\Psi,\hat{\Psi}}$ and the levels of the tree (sets of nodes of equal length). Let each strategy work on the requirement of its level. Also effectively associate each strategy with an infinite recursive set of integers $S_\alpha = \hat{S}_\alpha$ (such that $\bigsqcup_{\alpha \in T} S_\alpha = \omega$), and let α work with pairs $(i, \hat{\imath}) \in S_\alpha \times \hat{S}_\alpha$.

The A and $\hat{A}$-sides of a strategy α's full module proceed as described in Diagrams 1 and 2, respectively.

In general, parameters without hats refer to the A-side, parameters with hats refer to the $\hat{A}$-side of the module. We assume that γ, the use of Γ, is computed separately on A and W, so $\Gamma^{A \oplus W}(x, s) \downarrow$ implies $\Gamma^{A \restriction (\gamma(x,s)+1) \oplus W \restriction (\gamma(x,s)+1)}(x, s) \downarrow$.

The parameters i, n, r, and v_j^k (for $j = 0, 1$; $k \in \omega$) are defined in the flow chart and roughly denote the *candidate for an inequality* at which α is trying to establish $\lim_s \Gamma^{A \oplus W}(i, s) \neq \lim_v \Psi^A(i, v)$, the *number of the Ψ-flip* that α is trying to achieve now, the *A-restraint α imposes*, and the *opponent's "stage" v* at which he establishes $\Psi^A(i, v) = j$ for the kth time. The current stage is denoted by s. To *initialize* α means to put both sides into *init* and to set the restraints to zero, to *initialize the $\hat{A}$-side* means to do this for the $\hat{A}$-side only.

The following parameters referred to in the diagrams are defined in the text:

The A-side respects A-restraint $r' = \max\{\, r(\beta) \mid \beta^\frown\langle \hat{\mathit{flip}}\rangle \subseteq \alpha \,\}$, the A-restraint imposed by strategies that α assumes will get finitely many Ψ-flips and infinitely many $\hat{\Psi}$-flips. Note that α can afford to do so since it assumes that r' has a finite limit on the set of stages when it acts.

To organize the delay properly, the module defines $P(\alpha)$ and $\hat{P}(\alpha)$ (whenever indicated in the diagram) by setting it to a number greater than all current values of $P(\tilde{\alpha})$ (or $\hat{P}(\tilde{\alpha})$) for any $\tilde{\alpha} \in T$. Intuitively, α cannot injure the first $P(\alpha)$ many Ψ-flips or the first $\hat{P}(\alpha)$ many $\hat{\Psi}$-flips of stronger strategies.

We define $u(n)$ (the assigned use for $\gamma(i, s_1)$) to be the least $y \in S_\alpha$ greater than all of the following:

(i) $\psi(i, v_0^n)$;
(ii) all previous values of the parameter $\gamma(i, s_1)$;
(iii) $\max\{\, r(\beta) \mid \beta^\frown\langle \mathit{flip}\rangle <_L \alpha \,\}$; and
(iv) $\psi_\beta(i_\beta, v_{1,\beta}^{P(\alpha)})$ for all β with $\beta^\frown\langle \mathit{flip}\rangle \subseteq \alpha$.

(Here, $r(\beta)$ is the A-restraint imposed by β. Notice that for β with $\beta^\frown\langle \mathit{flip}\rangle \subseteq \alpha$, α observes only the part of the A-restraint imposed by β that it is not allowed to injure.)

Likewise, $\hat{u}(\hat{n})$ (the assigned use for $\hat{\gamma}(\hat{\imath}, \hat{s}_1)$) is the least $y \in \hat{S}_\alpha$ greater than all of the following:

(i) $\hat{\psi}(\hat{\imath}, \hat{v}_0^{\hat{n}})$;
(ii) all previous values of the parameter $\hat{\gamma}(\hat{\imath}, \hat{s}_1)$;
(iii) $\hat{r}(\beta)$ for all *be* with $\beta^\frown\langle \hat{\mathit{flip}}\rangle <_L \alpha$ associated with the same Ψ (and

thus $\hat{A}$); and

(iv) $\hat{\psi}_\beta(\hat{\imath}_\beta, \hat{v}_{1,\beta}^{\hat{P}(\alpha)})$ for all β with $\beta^\frown\langle \hat{flip}\rangle \subseteq \alpha$ associated with the same Ψ (and thus $\hat{A}$).

(Notice that we will have $\hat{r}(\beta) = 0$ for β with $\beta^\frown\langle flip\rangle \subseteq \alpha$ whenever α acts since β's $\hat{A}$-side will just have been initialized, so α need not consider the $\hat{A}$-restraint of these β.)

This ends the description of the full module of an individual strategy. We will now describe the strategy control.

At stage 0, the strategy control will set all parameters to 0 or $\emptyset$ (except $\Gamma^{A\oplus W}(x, s)$ and $\gamma(x, s)$ for $s > 0$ and their counterparts with hats).

At each stage $s > 0$, the strategy control will perform the following three steps:

1) It will let each strategy α whose A-side (or $\hat{A}$-side) is in *hold* (or *hôld*) go to *Wchange* (or *Wchânge*) and on to the next state if $W_s \upharpoonright (\gamma(i, s_1)+1) \neq W_{s_2} \upharpoonright (\gamma(i, s_1)+1)$ (or $W_s \upharpoonright (\hat{\gamma}(\hat{\imath}, \hat{s}_1)+1) \neq W_{\hat{s}_2} \upharpoonright (\hat{\gamma}(\hat{\imath}, \hat{s}_1)+1)$, respectively). (Notice that this action does not interfere with any other strategies.)

2) At each substage $t \leq s$ of stage s, some strategy α (with $|\alpha| = t$) will be *eligible to act*. Strategy $\emptyset$ will be eligible to act at substage 0; if α acted at substage t, then $\alpha^\frown\langle \mathrm{a}\rangle$ will be eligible to act at substage $t+1$ where a is the temporary outcome of α (as defined below).

3) At the end of stage s, the strategy control will define $\Gamma^{A\oplus W}(x, s')$ (and all $\hat{\Gamma}^{\hat{A}\oplus W}(x, s')$) for all $x \in \omega$ and all $s' \leq s$ to ensure that $\Gamma^{A\oplus W}$ and $\hat{\Gamma}^{\hat{A}\oplus W}$ are total (as outlined before the description of the basic module).

The rest of this section is devoted to describing in detail the action at substage t under step 2. At each substage t, the strategy control will first check if the strategy α that is eligible to act is delayed or postponed. α is *delayed on the A-side* if there is some β with $\beta^\frown\langle flip\rangle \subseteq \alpha$ such that

$n(\beta) \leq P(\alpha)$ where $n(\beta)$ is β's parameter n (the number of the Ψ-flip that β is trying to achieve now). Likewise, α is *delayed on the $\hat{A}$-side* if there is some β with $\beta^\frown\langle \hat{flip}\rangle \subseteq \alpha$ and associated with the same Ψ (and thus $\hat{A}$) such that $\hat{n}(\beta) \leq \hat{P}(\alpha)$ where $\hat{n}(\beta)$ is defined analogously. α is *postponed on the A-side* if there is some β with $\beta^\frown\langle flip\rangle \subseteq \alpha$ or $\beta^\frown\langle \hat{flip}\rangle \subseteq \alpha$ such that if α acted now it would measure (in a decision), or restrain, A at or above $\gamma(i(\beta), s_1(\beta))$ (and thus might be injured later by this β). Likewise, α is *postponed on the $\hat{A}$-side* if there is some β with $\beta^\frown\langle flip\rangle \subseteq \alpha$ or $\beta^\frown\langle \hat{flip}\rangle \subseteq \alpha$ associated with the same Ψ (and thus $\hat{A}$) such that if α acted now it would measure or restrain $\hat{A}$ at or above $\hat{\gamma}(\hat{i}(\beta), \hat{s}_1(\beta))$.

If α is delayed or postponed then the strategy control will initialize all $\beta >_L \alpha$ and start the next substage with $\alpha^\frown\langle fin\rangle$. Otherwise, we let α act according to the flow chart on the A-side if that side is not in *hold*; and on the $\hat{A}$-side otherwise. (Notice that only one side of α will act unless the flow chart explicitly starts up the action on the other side in which case both sides will act.)

If there is some β with $\beta^\frown\langle flip\rangle \subseteq \alpha$ and α puts some $x \leq r(\beta)$ into A, then β has been *injured explicitly* by α on the A-side as x's entering A changes an A-computation that β was preserving. In this case, each such β will perform *injury action on the A-side* as follows:

(i) if now $\Gamma^{A\oplus W}(i, s_1)\downarrow = 1$ then β goes to *inj* and on to the next state;

(ii) otherwise, β goes to inj_{m_β} where $m_\beta = \min\{\, m \mid x \leq \psi_\beta(i_\beta, v^m_{1,\beta})\,\}$ (the number of the least injured Ψ-flip) and on to the next state.

(In the first case, we have $x > \gamma(i, s_1)$, and so $\Gamma^{A\oplus W}(i, s_1)$ cannot (and need not) be reset to 0. In this case, β's A-side must be in *hold*, and so β just gives up the first half of the Ψ-flip it is currently trying to achieve. In the second case, β goes back to the situation before it established the

mth Ψ-flip.)

Likewise, if there is some β with $\beta^\frown\langle \hat{flip}\rangle \subseteq \alpha$ associated with the same Ψ and (thus $\hat{A}$) such that α put some $x \leq \hat{r}(\beta)$ into $\hat{A}$, then β has been *injured explicitly on the $\hat{A}$-side*, and we let β perform the corresponding *injury action on the $\hat{A}$-side* (using the counterparts of the above with hats).

Furthermore, the strategy control determines the *temporary outcome* a of α. It will be:

(i) *flip*, if the A-side of α went from *flip* to *wait0*;

(ii) *fl̂ip*, if the A-side of α went from *hold* to *wait0* and, since the last time the A-side was in *hold*, the $\hat{A}$-side went from *fl̂ip* to *wait0̂* and has not been initialized or injured since (this is the time when α's A-restraint is low); and

(iii) *fin*, otherwise.

Finally, the strategy control will initialize all $\gamma >_L \alpha^\frown\langle \mathrm{a}\rangle$; if either side of α changed states, it will also initialize all $\gamma \supseteq \alpha^\frown\langle fin\rangle$.

The Verification. Let δ_s, the *recursive approximation to the true path*, be the string of strategies that are eligible to act at stage s. Let $f = \liminf_s \delta_s$ be the *true path*, and let $f_0 = \bigcup\{\alpha \in f \mid \alpha \text{ initialized at most finitely often}\}$ be the *correct part of the true path* (which is possibly only a finite initial segment of f). Intuitively, f_0 will be finite if we discover at that finite level of the tree that W is recursive. Otherwise, $f = f_0$.

LEMMA 1 (INJURY FROM BELOW LEMMA). *If $\alpha < \beta$ then at any stage s, β injures α only* explicitly *(i.e., β injures α only if $\alpha^\frown\langle flip\rangle \subseteq \beta$ or $\alpha^\frown\langle \hat{flip}\rangle \subseteq \beta$, and α performs injury action), and β does not injure α's first $P(\alpha)$ (or $\hat{P}(\alpha)$) many Ψ-flips (or $\hat{\Psi}$-flips).*

PROOF: β can only injure α on the A-side at stage s if $\beta \subseteq \delta_s$, i.e., if β acts at stage s. At that stage, β will put its $\gamma(i, s_1)$ into A. This $\gamma(i, s_1)$ was defined at stage $s_1 < s$, and at that time $\gamma(i, s_1) > r_{s_1}(\alpha)$ if $\alpha^\frown\langle \mathit{flip}\rangle <_L \beta$, and $\gamma(i, s_1) > \psi_\alpha(i_\alpha, v_{1,\alpha}^{P(\beta)})[s_1]$ if $\alpha^\frown\langle \mathit{flip}\rangle \subseteq \beta$.

In the first case, α has increased its restraint since stage s_1, say, at stage s'. If $\alpha <_L \beta$ or $\alpha^\frown\langle \mathit{fin}\rangle \subseteq \beta$ then β was initialized and $\gamma(i, s_1)$ was cancelled at stage s'. If $\alpha^\frown\langle \widehat{\mathit{flip}}\rangle \subseteq \beta$ then β explicitly respects α's restraint.

On the other hand, if $\alpha^\frown\langle \mathit{flip}\rangle \subseteq \beta$ then α will perform injury action if β injures α. Furthermore, at stage s_1, $\gamma_\beta(i_\beta, s_1)[s_1] > \psi_\alpha(i_\alpha, v_{1,\alpha}^{P(\beta)})[s_1]$. Now, no strategy $\tilde{\beta}$ with $\tilde{\beta} <_L \beta$ or $\tilde{\beta}^\frown\langle \mathit{fin}\rangle \subseteq \beta$ can injure α's first $P(\beta)$ many Ψ-flips without β being initialized. If some $\tilde{\beta}$ with $\tilde{\beta}^\frown\langle \mathit{flip}\rangle \subseteq \beta$ or $\tilde{\beta}^\frown\langle \widehat{\mathit{flip}}\rangle \subseteq \beta$ injures α's first $P(\beta)$ many Ψ-flips then $\tilde{\beta}$ puts its $\gamma_{\tilde{\beta}}(i_{\tilde{\beta}}, s_{1,\tilde{\beta}})$ into A. But then β would have been postponed with defining its $\gamma(i, s_1)$ until after $\tilde{\beta}$'s injury to α. Any $\tilde{\beta}$ with $\tilde{\beta} >_L \beta$ or $\tilde{\beta} \supseteq \beta^\frown\langle \mathit{fin}\rangle$ is initialized at stage s_1 and therefore $P(\tilde{\beta}) > P(\beta)$ and $\tilde{\beta}$ cannot injure α's first $P(\beta)$ many Ψ-flips. No $\tilde{\beta}$ with $\tilde{\beta} \supseteq \beta^\frown\langle \mathit{flip}\rangle$ or $\tilde{\beta} \supseteq \beta^\frown\langle \widehat{\mathit{flip}}\rangle$ can injure α's first $P(\beta)$ many Ψ-flips between stage s_1 and stage s, or else it would also injure β, and $\gamma_\beta(i_\beta, s_{1,\beta})$ would be redefined. Therefore, $\psi_\alpha(i_\alpha, v_{1,\alpha}^{P(\beta)})[s_1] = \psi_\alpha(i_\alpha, v_{1,\alpha}^{P(\beta)})[s]$, and so β will not injure α's first $P(\beta)$ many flips.

The proof for the $\hat{A}$-side is the same except that we note that β cannot injure α on the $\hat{A}$-side if $\alpha^\frown\langle \mathit{flip}\rangle \subseteq \beta$ since the $\hat{A}$-side of α has just been initialized and α's $\hat{A}$-restraint is zero whenever β acts. $\diamond$

LEMMA 2 (INJURY FROM ABOVE LEMMA). *If $\alpha \subset \beta$ then at any stage s, α will not injure β by putting $x \leq r(\beta)$ into A or $x \leq \hat{r}(\beta)$ into $\hat{A}$.*

PROOF: Note that any $\beta \supseteq \alpha^\frown\langle fin\rangle$ will be initialized if α puts any number into A or $\hat{A}$. If $\beta \supseteq \alpha^\frown\langle flip\rangle$ or $\beta \supseteq \alpha^\frown\langle f\hat{l}ip\rangle$ then β will be postponed until α cannot injure it. $\diamond$

Notice the unusual feature that for $\alpha \subset \beta$, the weaker β may injure the stronger α infinitely often (in a controlled way), but that β is too smart to be injured by α.

LEMMA 3 (NUMBER OF FLIPS LEMMA). *If $\alpha \subseteq f_0$ and $\alpha^\frown\langle flip\rangle \subset f$ then* $\lim_s n(\alpha) = \infty$. *If $\alpha \subseteq f_0$ and $\alpha^\frown\langle f\hat{l}ip\rangle \subset f$ then* $\lim_s \hat{n}(\alpha) = \infty$ *and* $\lim_s n(\alpha) < \infty$ *exists.*

PROOF: Assume that α is never initialized after stage s'. Then $n(\alpha)$ is increased each time $\alpha^\frown\langle flip\rangle \subseteq \delta_s$. Furthermore, for each n, $n(\alpha)$ can be decreased to n through explicit injury only a finite number of times by Lemma 1 and the fact that the $P(\beta)$ increase. Therefore, $\lim_s n_s(\alpha) = \infty$.

The analogous proof shows that $\lim_s \hat{n}(\alpha) = \infty$ if we also assume that $\alpha^\frown\langle f\hat{l}ip\rangle \leq \delta_s$ for all $s > s'$ since the $\hat{A}$-side of α goes from $f\hat{l}ip$ to $wait\hat{0}$ infinitely often and is not initialized after stage s'. On the other hand, $n(\alpha)$ can only decrease after stage s' (or else we would have $\alpha^\frown\langle flip\rangle \subseteq \delta_s$ for some $s > s'$), so $\lim_s n(\alpha) < \infty$ exists. $\diamond$

The fact that strategies are allowed to injure higher-priority strategies infinitely often seems to prevent A from being low.

LEMMA 4 (DELAY/POSTPONEMENT LEMMA). *If $\alpha \subset f$ and both Ψ^A and $\hat{\Psi}^{\hat{A}}$ are total, then α is not delayed or postponed at cofinitely many α-stages (stages such that $\alpha \subseteq \delta_s$).*

PROOF: Suppose for the sake of contradiction that α is always delayed or postponed at α-stages after some stage s', say. Now any delay is finite since $\lim_s n(\beta) = \infty$ ($\lim_s \hat{n}(\beta) = \infty$) for each β with $\beta^\frown\langle flip\rangle \subseteq \alpha$ ($\beta^\frown\langle f\hat{l}ip\rangle \subseteq \alpha$, respectively) by Lemma 2, but $P(\alpha)$ (or $\hat{P}(\alpha)$) is constant after stage s'. So suppose α is always postponed after stage $s'' \geq s'$, say. Since Ψ^A and $\hat{\Psi}^{\hat{A}}$ are total, their uses settle down. Moreover, the A-restraint of any β with $\beta^\frown\langle \mathit{flip}\rangle \subseteq \alpha$ (and the $\hat{A}$-restraint of any β with $\beta^\frown\langle \hat{\mathit{flip}}\rangle \subseteq \alpha$ associated with the same Ψ (and thus $\hat{A}$)) settles down on the α-stages. (For the A-restraint of such a β use the fact that α is eligible to act only when β's A-restraint is down by the definition of the temporary outcome of β.) Finally, $n(\beta)$ (and $\hat{n}(\beta)$) tends to infinity for these β, so $\psi_\beta(i_\beta, v_{1,\beta}^{P(\alpha)})$ (and $\hat{\psi}_\beta(\hat{\imath}_\beta, \hat{v}_{1,\beta}^{\hat{P}(\alpha)})$) settles down for these β. Therefore, $u(n)$ (and $\hat{u}(\hat{n})$) settles down. But $\gamma(i(\beta), s_1(\beta))$ (and $\hat{\gamma}(\hat{\imath}(\beta), \hat{s}_1(\beta))$) tends to infinity for any such β, so α will not be postponed eventually. $\diamond$

LEMMA 5 (CONVERGENCE LEMMA). *(i)* $\Gamma^{A\oplus W}$ *is total, and for all* x, $\lim_s \Gamma^{A\oplus W}(x,s)$ *exists.*

(ii) For all Ψ, $\hat{\Gamma}_\Psi^{\hat{A}_\Psi \oplus W}$ *is total, and for all* x, $\lim_s \hat{\Gamma}_\Psi^{\hat{A}_\Psi \oplus W}(x,s)$ *exists.*

PROOF: It follows immediately from the construction (step 3) that $\Gamma^{A\oplus W}$ and all of the $\hat{\Gamma}_\Psi^{\hat{A}_\Psi \oplus W}$ are total. All $\hat{\Gamma}_\Psi^{\hat{A}_\Psi \oplus W}$ have limits since we ensure $\hat{\Gamma}_\Psi^{\hat{A}_\Psi \oplus W}(x,s) \leq \hat{\Gamma}_\Psi^{\hat{A}_\Psi \oplus W}(x,s+1) \leq 1$. The same is almost true for $\Gamma^{A\oplus W}$ as well, except that some strategy α may not be able to reset a computation $\Gamma^{A\oplus W}(i,s) = 1$ on going from *hold* to *wait0* if $\gamma(i,s_1) \leq r'$. But for all β with $\beta^\frown\langle \hat{\mathit{flip}}\rangle \subseteq \alpha$, $\lim_s n(\beta) < \infty$ exists (by Lemma 3), and thus so does $\lim_{s\in S^\alpha} r(\beta) < \infty$ where $S^\alpha = \{\, t \mid \alpha \subseteq \delta_t \,\}$. So $\lim \Gamma^{A\oplus W}(i,s)$ also exists for those i. $\diamond$

We now analyze the outcomes:

LEMMA 6 (FINITE OUTCOME LEMMA). *Suppose* $\alpha \subseteq f_0$, *both* Ψ^A *and* $\hat{\Psi}^{\hat{A}}$ *are total, and eventually neither the* A*-side nor the* $\hat{A}$*-side changes states. Then:*

(i) eventually, the A-side of α stays in wait0 or wait1, or the $\hat{A}$-side stays in wait$\hat{0}$ or wait$\hat{1}$; and

(ii) either not $\lim_v \Psi^A(i,v) = \lim_s \Gamma^{A\oplus W}(i,s)$ *or not* $\lim_v \hat{\Psi}^{\hat{A}}(\hat{\imath},v) = \lim_s \hat{\Gamma}^{\hat{A}\oplus W}(\hat{\imath},s)$ *for the eventual candidates i and $\hat{\imath}$ of α.*

PROOF: (i) By the construction and Lemma 4, the A-side can get stuck only in *wait0*, *wait1*, or *hold*. If the A-side is stuck in *hold* then the $\hat{A}$-side must be stuck in *wait*$\hat{0}$ or *wait*$\hat{1}$.

(ii) Otherwise, α would leave the states mentioned in (i) by the construction and Lemma 4. ◇

LEMMA 7 (FLIP OUTCOMES LEMMA). *(i) If $\alpha \subseteq f_0$ and $\alpha^\frown\langle flip\rangle \subset f$ then* $\lim_v \Psi^A(i,v)$ *does not exist for the eventual candidate i of α.*

(ii) If $\alpha \subseteq f_0$ and $\alpha^\frown\langle fl\hat{\imath}p\rangle \subset f$ then $\lim_v \hat{\Psi}^{\hat{A}}(\hat{\imath},v)$ *does not exist for the eventual candidate $\hat{\imath}$ of α.*

PROOF: By the construction, the candidate i ($\hat{\imath}$) settles down in case (i) (case (ii), respectively), and by Lemma 2, $n(\alpha)$ ($\hat{n}(\alpha)$) tends to infinity. But $n(\alpha)-1$ ($\hat{n}(\alpha)-1$) is the number of protected flips from 0 to 1 back to 0 of $\Psi^A(i,-)$ ($\hat{\Psi}^{\hat{A}}(\hat{\imath},-)$), so the limit of Ψ ($\hat{\Psi}$) cannot exist. ◇

LEMMA 8 (RECURSIVE OUTCOME LEMMA). *If $\alpha = f_0$ is of finite length, then W is recursive.*

PROOF: First of all, $\alpha^\frown\langle flip\rangle \subset f$ or $\alpha^\frown\langle fl\hat{\imath}p\rangle \subset f$ is impossible by the way the initialization is arranged; thus $\alpha^\frown\langle fin\rangle \subset f$. So suppose that $\alpha^\frown\langle fin\rangle \leq \delta_s$ for all $s > s'$, say. Thus $n(\alpha)$ and $\hat{n}(\alpha)$ eventually come to a finite limit, and by Lemmas 1 and 2, α is never injured after stage s'. Since $\alpha^\frown\langle fin\rangle$ is initialized infinitely often, α keeps changing states. Both sides settle down on candidates i and $\hat{\imath}$ after stage $s'' \geq s'$, $\lim_s \gamma(i,s) = \lim_s \hat{\gamma}(\hat{\imath},s) = \infty$, and both these parameters are nondecreasing in s. Also, after stage s'', both sides always destroy their Γ- and $\hat{\Gamma}$-computations, and thus $W \upharpoonright \gamma(i, s_1)$ does not change while the A-side is in *hold*, and $W \upharpoonright \hat{\gamma}(\hat{\imath}, \hat{s}_1)$ does not change while the $\hat{A}$-side is in *hôld*. Therefore, W is recursive. ◇

Lemma 8 immediately yields Lemma 9:

LEMMA 9 (INFINITE TRUE PATH LEMMA). *If W is not recursive then f_0 is infinite.* ◇

Thus, if W is not recursive, then $\alpha \subset f_0$ of each level will satisfy its requirement by Lemmas 6 and 7. This concludes the proof of the theorem. ◇

§4. A "MODERATELY" DEEP DEGREE

In the construction of the previous section, the jump of A is not controlled. Due to the infinitary outcomes of the strategies influencing A's construction there does not seem to be an obvious way to make A low whenever W is nonrecursive. In fact, it seems quite conceivable to the authors that for some nonrecursive *low* recursively enumerable degree $\mathbf{w}$, $\mathbf{a} \cup \mathbf{w}$ is low for any low recursively enumerable degree $\mathbf{a}$[2]. In the following, we will prove a weaker version of this.

Jockusch (private communication) raised the question whether there is a nonrecursive low recursively enumerable degree that does not join with any other low recursively enumerable degree to a high degree. We answer this question positively (reversing the roles of $\mathbf{a}$ and $\mathbf{w}$ conforming with our convention on names of objects built by us or built by the opponent):

[2] Harrington (unpublished) has recently shown this.

THEOREM. *There is a low recursively enumerable degree* $\mathbf{a} \neq \mathbf{0}$ *such that for all low recursively enumerable degrees* $\mathbf{w}$, $\mathbf{a} \cup \mathbf{w}$ *is not high.*

PROOF: We will drop the restriction that $\mathbf{a}$ be low, since if $\mathbf{a}$ is not low choose $\mathbf{a}_0 < \mathbf{a}$ low which satisfies the theorem. (However, a closer analysis shows that our $\mathbf{a}$ is already low.)

We have, for all recursively enumerable sets W_e, the usual positive requirements for nonrecursiveness:

$$(7) \qquad \mathcal{P}_e : \overline{A} \neq W_e,$$

and, for all recursively enumerable sets W, the requirements:

$$(8) \qquad \hat{\mathcal{R}}_W : W \text{ nonlow or } W \oplus A \text{ nonhigh.}$$

The Strategy. We have to construct a recursively enumerable set A satisfying all requirements.

The opponent will try to put up a recursively enumerable set W and a functional Φ claiming that W is low and $\Phi^{W \oplus A}$ is total and dominates all total recursive functions, and thus, by a theorem of Martin [**Ma66**], $W \oplus A$ is high.

We will respond by trying to build a functional $\Gamma_{(W,\Phi)}$ witnessing the nonlowness of W via $\lim_s \Gamma^W_{(W,\Phi)}(-,s) \not\leq_T \emptyset'$.

If the opponent succeeds in refuting this by furnishing some total recursive function Ψ such that $\lim_s \Gamma^W_{(W,\Phi)}(-,s) = \lim_v \Psi(-,v) \leq_T \emptyset'$ then we will defeat him by constructing a total recursive function $\Delta_{(W,\Phi,\Psi)}$ that is not dominated by $\Phi^{W \oplus A}$. (We will use $\Delta_{(W,\Phi,\Psi)}$ to try to force changes in W to redefine $\Gamma^W_{(W,\Phi)}$.)

The requirements are thus of the form

$$(9)\quad \mathcal{R}_{W,\Phi,\Psi} : \Phi^{W\oplus A} \text{ total } \implies \Big(\lim_s \Gamma^W_{(W,\Phi)}(-,s) \neq \lim_v \Psi(-,v) \vee \\ \big[\Delta_{(W,\Phi,\Psi)} \text{ total} \,\&\, (\exists^\infty j)[\Delta_{(W,\Phi,\Psi)}(j) > \Phi^{W\oplus A}(j)]\big]\Big).$$

Now fix W and Φ and assume that $\Phi^{W\oplus A}$ is total. Then we will either satisfy $\mathcal{R}_{W,\Phi,\Psi}$ for all Ψ by the first disjunct, and thus W is nonlow; or we will satisfy one $\mathcal{R}_{W,\Phi,\Psi}$ by the second disjunct, and therefore $W \oplus A$ is not high via Φ. (We will suppress the subscripts on Γ and Δ if they are clear from the context.) We assume that φ, the use of Φ, is computed separately on W and A, so $\Phi^{W\oplus A}(x)\downarrow$ implies $\Phi^{W\restriction(\varphi(x)+1)\oplus A\restriction(\varphi(x)+1)}(x)\downarrow$.

The basic module for $\mathcal{R}_{W,\Phi,\Psi}$ consists of a stack of ω copies, each denoted by C_n, of a simple *submodule*. Copy C_0 acts first, each copy C_{n+1} is started by copy C_n, and a copy C_n can be initialized by a copy C_m with $m < n$.

Each copy will first try to show that $\lim_s \Gamma^W(i,s) \neq \lim_v \Psi(i,v)$ for its candidate i. If W won't let us reset $\Gamma^W(i,s)$, we have indirect restraint on W, which, combined with our restraint on A, helps us show that $\Phi^{W\oplus A}(j)$ is fixed and less than the $\Delta(j)$ defined by us.

Copy C_n thus proceeds as follows:

(i) pick a new candidate i (for $\lim_s \Gamma^W(i,s) \neq \lim_v \Psi(i,v)$),

(ii) pick the least j for which Δ is undefined,

(iii) start setting $\Gamma^W(i,s) = 0$ (until (iv) holds) at each stage s,

(iv) wait for $\Psi(i,v_0)\downarrow= 0$ for some v_0 and $\Phi^{W\oplus A}(j)\downarrow$ (at some stage s_1, say),

(v) impose A-restraint on $A\restriction(\varphi(j)+1)$,

(vi) start setting $\Gamma^W(i,s) = 1$ with $\gamma(i,s) = \varphi(j)$ (until (vii) or (x) holds) at each stage s,

(vii) if $W_s \upharpoonright (\varphi(j)+1) \neq W_{s_1} \upharpoonright (\varphi(j)+1)$ then immediately reset $\Gamma^W(i,s') = 0$ for $s_1 \leq s' \leq s$, cancel the A-restraint, and go to (iii),

(viii) wait for $\Psi(i, v_1) = 1$ for some $v_1 > v_0$ (at some stage s_2, say),

(ix) set $\Delta(j) > \Phi^{W \oplus A}(j)$ and start copy C_{n+1} (with different i and j),

(Notice that we now have a squeeze on W. If W changes we can reset our Γ while his Ψ has a flip; if W does not change we have another witness j towards showing that $\Phi^{W \oplus A}$ does not dominate Δ.)

(x) if $W_s \upharpoonright (\varphi(j)+1) \neq W_{s_1} \upharpoonright (\varphi(j)+1)$ then initialize copies C_m (for $m > n$), reset $\Gamma^W(i,s') = 0$ for $s_1 \leq s' \leq s$, cancel the A-restraint, and go to (ii) (looking for a new v_0 greater than the current v_1).

Here, all copies work on the same A, Γ, and Δ.

To ensure that Γ is total and that the limits exist, we use the same convention as in the previous section (described just before the basic module). We always pick the least j for which Δ is undefined in order to ensure that Δ is total if we pick infinitely many j.

Let $n_0 = \liminf_s \{\, n \mid \text{copy } C_n \text{ waiting for (iv) or (viii) at stage } s \,\}$ (possibly $n_0 = \infty$). The possible outcomes of the basic module are as follows:

(a) $n_0 = \infty$: Then each time a copy acts for the last time, it finds some j such that $\Delta(j) > \Phi^{W \oplus A}(j) \downarrow$, and therefore there are infinitely many such j witnessing that $W \oplus A$ is not high via Φ.

(b) $n_0 < \infty$: We distinguish the following cases:

(b_1) copy C_{n_0} acts finitely often (and therefore so does the whole module): Then C_{n_0} gets stuck at (iv) or (viii), and it is not the case that $\lim_s \Gamma^W(i,s) = \lim_v \Psi(i,v)$.

(b$_2$) copy C_{n_0} goes infinitely often through (x): Then $\lim_v \Psi(i,v)$ does not exist where i is the eventual candidate of C_{n_0} since we force infinitely many Ψ-flips.

(b$_3$) copy C_{n_0} goes finitely often through (x), but infinitely often through (vii): Then $\Phi^{W\oplus A}(j)$ is not defined for the eventual candidate j of C_{n_0}, and therefore $\Phi^{W\oplus A}$ is not total.

There are two problems with putting this module on a tree. Firstly, the restraint tends to infinity under outcome (a). But most of all, the natural ordering for the outcomes would be of order type $\omega + 2$ (namely, (b$_2$) for $n_0 = 0$ $<$ (b$_3$) for $n_0 = 0$ $<$ (b$_2$) for $n_0 = 1$ $<$ (b$_3$) for $n_0 = 1$ $< \cdots <$ (a) $<$ (b$_1$)), which would be cumbersome to organize on a tree.

On the other hand, each positive strategy for $\mathcal{P}_e$ acts at most once, and each copy of the above module can live with finite injury. So we will spread out the copies as separate strategies without giving up their coordination described above. We will use a linear priority ranking of these strategies combined with the method of W-true stages and the "hat trick", so called because of its original notation. For this, we tacitly assume that when $\Phi^{W\oplus A}(x)[s-1]\downarrow$ and some $x \leq \varphi(x)[s-1]$ enters W or A at stage s, then $\Phi^{W\oplus A}(x)$ is undefined at stage s. Call s a W-true stage if some x enters W at stage s and $W_s \upharpoonright x = W \upharpoonright x$. Then the A-restraint of each copy of a strategy has a finite limit on W-true stages.

The construction will thus "look like" a finite injury argument. However, to figure how each requirement $\mathcal{R}_{W,\Phi,\Psi}$ became satisfied will require a $\mathbf{0}'''$-oracle; in fact, it has to since the question of whether $W \oplus A$ is high via dominating functional Φ is Π_4-complete and thus the way in which each $\mathcal{R}_{W,\Phi,\Psi}$ becomes satisfied constitutes a Σ_3-complete statement.

The Full Construction. Fix an effective 1–1 correspondence

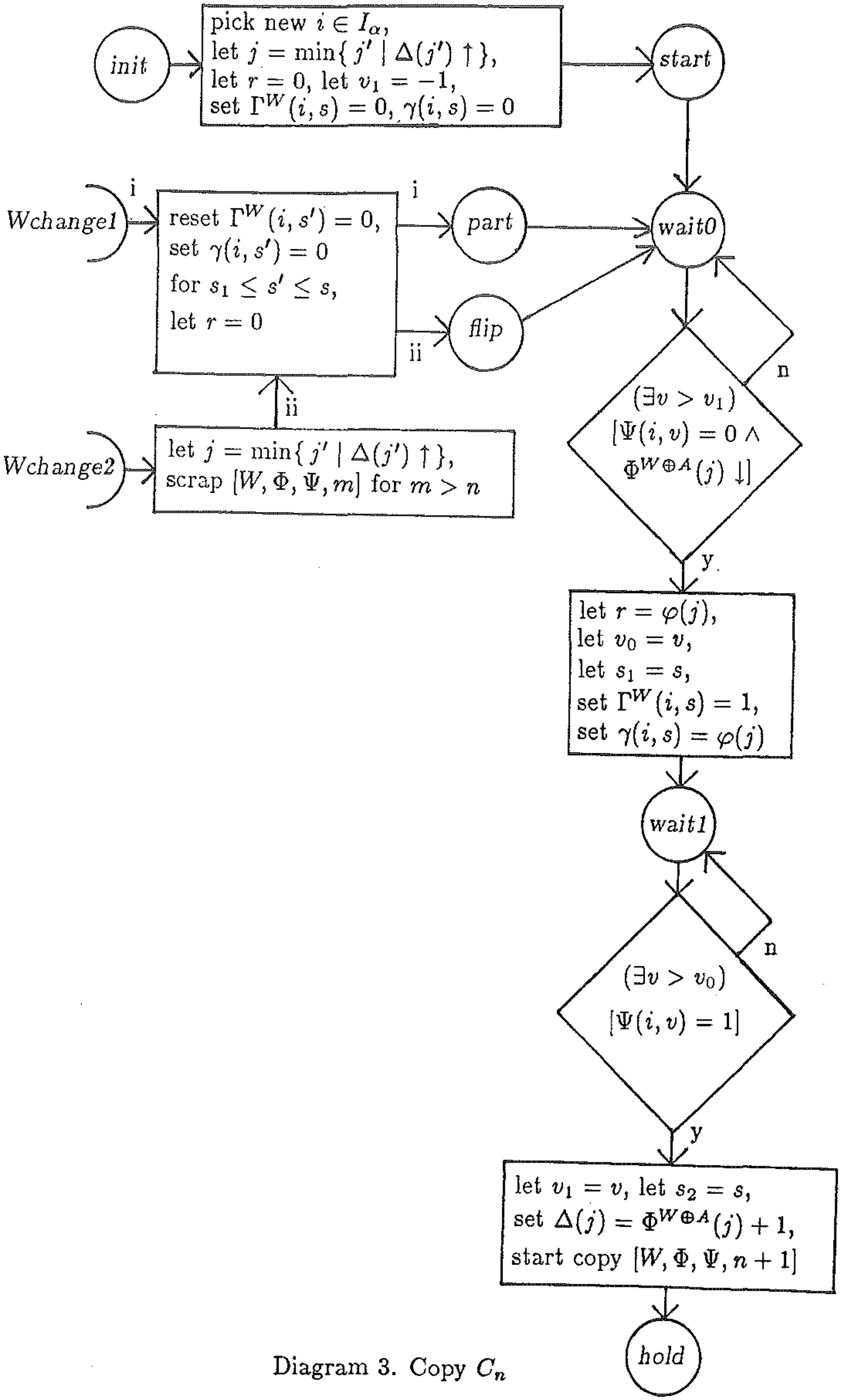

Diagram 3. Copy C_n

$\langle -,-,-,- \rangle$ between ω and all quadruples (W, Φ, Ψ, n) where W is a recursively enumerable set, Φ and Ψ are functionals, and n is an integer. (Assume here that always $\langle W, \Phi, \Psi, n \rangle < \langle W, \Phi, \Psi, n+1 \rangle$.) This correspondence will yield our priority ranking between strategies. We will denote a strategy α by $[W, \Phi, \Psi, n]$ to specify that α works as the nth copy in the basic module for $\mathcal{R}_{W,\Phi,\Psi}$.

The Γ will be common to all strategies with the same W and Φ, so fix effectively for each α an infinite recursive set of integers I_α such that

$$\bigsqcup_{\substack{\alpha = [W,\Phi,\Psi,n] \\ \text{for some } \Psi, n}} I_\alpha = \omega \tag{10}$$

for each pair (W, Φ). Δ will be common to all strategies working on the same $\mathcal{R}_{W,\Phi,\Psi}$. The Γ and the Δ are never discarded even when individual strategies are scrapped.

The module for a strategy $\alpha = [W, \Phi, \Psi, n]$ now acts as described in Diagram 3.

Here, v_0 and v_1 are the "stages" at which the opponent establishes $\Psi(i, v_j) = j$ (for $j = 0, 1$). The current stage is denoted by s. The *A-restraint* imposed by α is denoted by r. To *start* α means to let it go from *init* to *start*. To *scrap* it means to put it into *init* and to set its restraint to 0. To *initialize* $\alpha = [W, \Phi, \Psi, n]$ is to scrap all $[W, \Phi, \Psi, m]$ for $m \geq n$ and, if $n = 0$ or $[W, \Phi, \Psi, n-1]$ is in *hold*, to start α.

At stage 0, the strategy control lets $A_0 = \emptyset$, starts all strategies $[W, \Phi, \Psi, 0]$, and defines $\Gamma^W(x, 0) = 0$ with $\gamma(x, 0) = 0$ for all x and all W and Γ.

At a stage $s > 0$, the strategy control proceeds in three steps:

1) Action for $\mathcal{P}_e$: If there is some e such that

$$(11) \quad A_s \cap W_{e,s} = \emptyset \,\&(\exists x \in \omega^{[e]})[x > \max\{\, r(\alpha) \mid \#\alpha \leq e\,\} \,\&\, x \in W_{e,s}],$$

(where $r(\alpha)$ is the restraint imposed by α and $\#\alpha = \langle W, \Phi, \Psi, n\rangle$ is the code number of α) then for the least such e, put the least such x into A and initialize all α with $\#\alpha > e$.

2) For each triple $(W,\ \Phi,\ \Psi)$, do the following: First check whether there is a strategy $\alpha = [W, \Phi, \Psi, n]$ in *wait1* or *hold* such that $W_s \restriction \varphi(j) \neq W_{s_1} \restriction \varphi(j)$. If so let the least such α go from *Wchange1* or *Wchange2* to *wait0* (depending on whether α was in *wait1* or *hold*, respectively). Otherwise let the unique $[W, \Phi, \Psi, n]$ that is not in *init* or *hold* act according to the flow chart.

3) The strategy control (re)defines $\Gamma^W(x, s') = \Gamma^W(x, s'-1)$ for all W and Γ with same use for all x and all $s' \leq s$ for which Γ is now undefined.

The Verification. We proceed in several lemmas:

LEMMA 1 (CONVERGENCE LEMMA). *For each pair (W, Φ), $\Gamma^W_{(W,\Phi)}$ is total, and for all x, $\lim_s \Gamma^W_{(W,\Phi)}(x, s)$ exists.*

PROOF: $\Gamma^W(x, s)$ is defined at the end of each stage $s' \geq s$ (by step 3 of the construction). $\gamma(x, s)$ increases at most once, so W-changes can make $\Gamma^W(x, s)$ undefined at most finitely often. As for the limit, note that for all x, s, $\Gamma^W(x, s) \leq \Gamma^W(x, s+1) \leq 1$. (So $\lim_s \Gamma^W(-, s)$ is actually Σ^W_1.) $\diamondsuit$

LEMMA 2 (FINITE INJURY LEMMA). *Action is taken for each $\mathcal{P}_e$ at most once, and thus each α is scrapped at most finitely often under step 1 of the construction.* $\diamondsuit$

Define a stage $s > 0$ to be *W-true* if $W \upharpoonright x = W_s \upharpoonright x$ for some $x \in W_s - W_{s-1}$. Let T be the (infinite) set of W-true stages. Note that, by the hat trick,

$$(12) \quad \Phi^{W \oplus A}(x)[s] \downarrow \,\&\, s \in T \,\&\, A_s \upharpoonright (\varphi_s(x)+1) = A \upharpoonright (\varphi_s(x)+1) \implies \Phi^{W \oplus A}(x) \downarrow .$$

LEMMA 3 (FINITE RESTRAINT/$\mathcal{P}_e$-STRATEGY LEMMA). *For any strategy α, $\lim_{s \in T} r[s] < \infty$ exists. (Thus each $\mathcal{P}_e$ is satisfied.)*

PROOF: By Lemma 2, let $\alpha = [W, \Phi, \Psi, n]$ not be scrapped under step 1 of the construction after stage s', say. Suppose for some $s_0 \in T$ with $s_0 > s'$, $r[s_0] > 0$. Then $\Phi^{W \oplus A}(j(\beta)) \downarrow$ at all stages $s \geq s'$ for all $\beta = [W, \Phi, \Psi, m]$ with $m \leq n$ via W-correct computations, which are also A-correct by the A-restraint and our assumption on s'. Thus in this case $r[s_0] = \lim_s r[s] < \infty$ exists. Otherwise $\liminf_s r[s] = \lim_{s \in T} r[s] = 0$. ◇

Now we fix W, Φ, and Ψ and distinguish the four possible cases for the outcome of the strategies $[W, \Phi, \Psi, n]$.

LEMMA 4 (FINITE OUTCOME LEMMA). *Suppose there are only finitely many stages at which any of the strategies $[W, \Phi, \Psi, n]$ (for fixed W, Φ, and Ψ) changes states. Then it is not the case that $\lim_s \Gamma^W(-, s) = \lim_v \Psi(-, v)$.*

PROOF: Let n_0 be the unique n such that $\alpha = [W, \Phi, \Psi, n]$ is eventually not in *init* or *hold*. Then α must be stuck in *wait0* or *wait1*. Therefore not $\lim_s \Gamma^W(i, s) = \lim_v \Psi(i, v)$ for the eventual candidate i of α. ◇

LEMMA 5 (FLIP LEMMA). *Suppose that for some n, $\alpha = [W, \Phi, \Psi, n]$ is scrapped finitely often and goes through Wchange2 infinitely often. Then $\lim_v \Psi(i, v)$ does not exist for the eventual candidate i of α.*

PROOF: Let α not be scrapped after stage s', say. Then the parameters v_0 and v_1 increase to infinity, and each time they increase, $\Psi(i, v_0) = 0$ or $\Psi(i, v_1) = 1$ is established. $\diamondsuit$

LEMMA 6 (PARTIAL Φ LEMMA). *Suppose that for some n, $\alpha = [W, \Phi, \Psi, n]$ is scrapped finitely often and changes states infinitely often, but goes through Wchange2 only finitely often. Then $\Phi^{W \oplus A}$ is partial.*

PROOF: Suppose that α is not scrapped or goes through *Wchange2* after stage s', say. Then α from now on always goes through *Wchange1* with the same j, so $\Phi^{W \oplus A}(j) \uparrow$. $\diamondsuit$

LEMMA 7 (NONDOMINANCE LEMMA). *Suppose that, for fixed W, Φ, and Ψ, no single $[W, \Phi, \Psi, n]$ changes states infinitely often, but that there are infinitely many stages at which some $[W, \Phi, \Psi, n]$ changes states. Then Δ is total and not dominated by $\Phi^{W \oplus A}$. (Thus $W \oplus A$ is not high via Φ.)*

PROOF: First of all, Δ is total since we always pick the least j for which Δ is currently undefined and since each $[W, \Phi, \Psi, n]$ is eventually in *hold*. But each time some $[W, \Phi, \Psi, n]$ reaches *hold* for the last time, $\Delta(j) > \Phi^{W \oplus A}(j)$ is established for its current j, and this is preserved by the A-restraint. $\diamondsuit$

Lemmas 2 and 4 through 7 establish the theorem. $\diamondsuit$

REFERENCES

[**BMta**] M. Bickford and C.F. Mills, *Lowness properties of r.e. sets*, (to appear).

[**Jo85**] C. G. Jocksuch, *Genericity for recursively enumerable sets*, in "Recursion Theory Week, Lecture Notes in Mathematics 1141," Ebbinghaus, Müller and Sacks, eds., Springer-Verlag, Berlin, 1985, pp. 203-232.

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

[**Ma66**] D.A. Martin, *Classes of Recursively Enumerable Sets and Degrees of Unsolvability*, Zeitschrift f. math. Logik und Grundlagen d. Math. **12** (1966), 295–310.

[**Ms82**] W. Maass, *Recursively enumerable generic sets*, Jour. Sym. Logic **47** (1982), 809–823.

[**Sa63a**] G.E. Sacks, *On the degrees less than* $\mathbf{0}'$, Annals of Mathematics **108** (1963), 211–231.

[**Sa63b**] ________, *Recursive Enumerability and the Jump Operator*, Trans. Amer. Math. Soc. **108** (1963), 223–239.

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

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

[**So85**] R.I. Soare, *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).

[**SS82**] R.I. Soare and M. Stob, *Relative Recursive Enumerability*, in "Proceedings of the Herbrand Symposium Logic Colloquium '81," Ed. J. Stern, North Holland, Amsterdam, pp. 299–324.

Yale University, New Haven, CT 06520
University of Chicago, Chicago, IL 60637