PRIORITY ARGUMENTS IN COMPUTABILITY THEORY, MODEL THEORY, AND COMPLEXITY THEORY

Steffen Lempp

Department of Mathematics
University of Wisconsin
Madison, WI 53706–1388, USA

0. Introduction

These notes present various priority arguments in classical computability theory, effective model theory, and complexity theory in a uniform style.[1]

Our notation usually follows Soare (1986) with some exceptions.

We view Turing functionals as c.e. sets Φ of triples $\langle x, y, \sigma\rangle$, denoting that $\Phi(\sigma; x) \downarrow= y$. (Of course, we have to impose the obvious compatibility condition, namely, that if $\langle x, y, \sigma\rangle, \langle x, z, \tau\rangle \in \Phi$ where σ and τ are comparable then $y = z$. It is not hard to check that any given Turing program can be effectively transformed into a computable enumeration of such triples coding the corresponding partial computable Turing functional, and that conversely every such c.e. set of triples can be effectively transformed into a Turing program for the corresponding partial computable Turing functional.)

The *use* of a computation $\Phi(X; x)$ is the largest number *actually used* in that computation, i.e., in the notation of the previous paragraph, the use equals $|\sigma| - 1$. Allowing a minor abuse of notation in the case when the oracle is given as the join of two or more sets, we then let the use be the largest number used on any set involved in the join. (Thus $\Phi((X_1 \oplus X_2)\restriction(u+1); x)$ is defined iff $\Phi((X_1\restriction(u+1) \oplus (X_2\restriction(u+1)); x)$ is.) We denote the use of a computation $\Phi(X; x)$ by the corresponding lower-case letter (i.e., $\varphi(X; x)$, or simply $\varphi(x)$ when the oracle is clear from the context). We adopt the following

Use Conventions. For any use function $\varphi_s(x)$ of a given Turing functional (i.e.,

---

1991 *Mathematics Subject Classification.* 03D25 and others.

Key words and phrases. priority argument.

The author would like to thank Chris Alfeld, Argimiro Arratia-Quesada, Alfred Dolich, Evan Griffiths, James Hunter, Asher Kach, Tolya Karp, Tom Kent, Noah Kieserman, Moses Klein, Ben Marsh, Dan McGinn, Alex Raichev, Maciej Smuga-Otto, Stephen Tanner, Todd Turnidge, and Dejia Wang for stimulating in-class discussions and numerous corrections.

[1]These notes are currently in the process of being written and thus unfinished. Comments and corrections are appreciated.

Typeset by $\mathcal{AMS}$-TEX

given by the "opponent"), we assume for all x and s that

$$\varphi_s(x) \leq \varphi_{s+1}(x) \text{ (if } \varphi_s(x) < \infty \text{ and the oracle is computably enumerable),}$$
$$\varphi_s(x) < \varphi_s(x+1) \text{ (if } \varphi_s(x) < \infty\text{),}$$
$$x \leq \varphi_s(x) \leq s \text{ (if } \varphi_s(x) < \infty\text{), and}$$
$$\varphi_s(x) = \infty \text{ (whenever } \Phi(X; x) \text{ is undefined at stage } s\text{).}$$

We highlight here a fact about computations relative to a Δ_2-oracle (and so in particular relative to a computably enumerable oracle) which is used throughout these notes. Suppose a Turing computation $\Phi(W; x)$ is defined for some Turing functional Φ, some Δ_2-oracle W, and some argument x. Then this computation can be "approximated in a Δ_2-fashion", i.e., given computable approximations $\{\Phi_s\}_{s\in\omega}$ and $\{W_s\}_{s\in\omega}$ to Φ and W, respectively, $\Phi_s(W_s; x)$ will be defined and equal $\Phi(W; x)$ for almost all s. (On the other hand, if $\Phi(W; x)$ is undefined then its approximation may still be defined for infinitely many s although the use $\varphi_s(x)$ will tend to infinity.) Note that in many constructions, we will only "believe" computations given by the "opponent" if we have seen this same computation at at least two stages. (This will present no problem since true computations relative to a Δ_2-oracle will appear to be true at cofinitely many stages.)

We finally note another convention about the proofs in these lecture notes: While describing a construction, we usually leave off the stage subscripts, since when we refer to objects during a construction, we clearly mean the object as constructed up to the current stage, or as measured at the current stage. During the verification, however, we have to refer to objects both in their final form and in their form approximated at a stage, so we will always use stage subscripts then.

1. Finite Injury or Π_1-Constructions

1.1. The Friedberg-Muchnik Theorem. The first priority argument was a theorem by Muchnik (1956) and Friedberg (1957), showing the existence of incomparable computably enumerable Turing degrees.

Theorem 1. (Muchnik (1956), Friedberg (1957)) *There are two incomparable computably enumerable Turing degrees* $\mathbf{a}_0$ *and* $\mathbf{a}_1$. *(Thus the computably enumerable degrees are not linearly ordered and consist of more than two degrees, solving Post's Problem.)*

Proof. We need to construct two computably enumerable sets A_0 and A_1, meeting, for all $i \leq 1$ and all Turing functionals Φ, the following

Requirements:

$$\mathcal{P}^i_\Phi : A_i \neq \Phi(A_{1-i}).$$

Strategy for $\mathcal{P}^i_\Phi$:

1. Pick an unused witness x (targeted for A_i) larger than any number mentioned so far in the construction and keep x out of A_i.
2. Wait for $\Phi(A_{1-i}; x) = 0$ (as measured at the current stage).
3. Enumerate x into A_i, preserve $\Phi(A_{1-i}; x) = 0$ by restraining numbers from entering $A_{1-i}\restriction(\varphi(x) + 1)$, and stop.

Outcomes:

w: Wait at Step 2 forever: Then $A_i(x) = 0 \neq \Phi(A_{1-i}; x)$.

s: Stop at Step 3: Then $A_i(x) = 1 \neq 0 = \Phi(A_{1-i}; x)$.

We let $\Lambda = \{s, w\}$ be the set of outcomes.

We distinguish two kinds of outcomes: The *current outcome* at the end of stage s is the outcome measured at the end of stage s, so here it is w if the strategy is still waiting at Step 2 by the end of stage s, and s otherwise. The *true outcome* is the outcome measured over all stages, so here it is w if the strategy waits at Step 2 forever, and s otherwise. Now, as in all finite injury priority arguments, the true outcome of the strategy is the current outcome of the strategy at cofinitely many stages.

Tree of strategies: Let $T = \Lambda^{<\omega}$ be the tree of strategies. Intuitively, this means that strategy $\alpha \in T$ assumes that $\beta \subset \alpha$ has true outcome o iff $\beta\,\hat{}\,\langle o \rangle \subseteq \alpha$. Effectively order the requirements (of order type ω) and assign the eth requirement in this list to all nodes ("strategies") $\alpha \in T$ of length e.

Construction: Each stage s of the construction consists of substages $t \leq s$. At substage t, a strategy $\alpha \in T$ with $|\alpha| = t$ is eligible to act and acts according to where it previously left off in the above description, i.e., α starts at Step 1 if it has never been eligible to act before; it continues from Step 2 if it was waiting at that step before; and it does nothing if it had stopped before. The strategy to be eligible to act at substage $t + 1$ is then $\alpha\,\hat{}\,\langle o \rangle$ where o is the current outcome of α at the end of substage t of stage s.

Verification: Denote by $f_s \in T$ the current true path at stage s, i.e., the longest strategy eligible to act at stage s. Denote by $f \in [T]$ the true path of the construction, i.e., the limit of f_s as s tends to infinity. (Note that f must exist since the outcome of each individual node eventually reaches a limit.)

Lemma. *Every strategy $\alpha \subset f$ ensures the satisfaction of its requirement.*

Proof. Fix $s_0 \geq |\alpha|$ least such that $\alpha \subseteq f_s$ for all $s \geq s_0$. Then α picks a witness x at stage s_0.

Case 1: α waits at Step 2 forever. Then $A_i(x) = 0 \neq \Phi(A_{1-i}; x)$.

Case 2: α eventually stops at Step 3 at substage $t_1 = |\alpha|$ of a stage $s_1 \geq s_0$, say. Then $A_i(x) = 1 \neq 0 = \Phi(A_{1-i}; x)[s_1]$. It remains to see that no strategy β can enumerate a number $\leq \varphi(x)$ into A_{1-i} after substage t_1 of stage s_1. For β incomparable with $\alpha\,\hat{}\,\langle s \rangle$, this holds since β is never eligible to act after substage t_1 of stage s_1. For $\beta \subset \alpha$, this holds since β cannot change outcome after substage t_1 of stage s_1 as $\alpha\,\hat{}\,\langle s \rangle \subseteq f_s$ for all $s \geq s_1$. For $\beta = \alpha$, β does not enumerate into A_{1-i}. And for $\beta \supseteq \alpha\,\hat{}\,\langle s \rangle$, this holds since β cannot choose a witness before substage t_1 of stage s_1, and so β's witness must be chosen greater than α's A_{1-i}-restraint.

1.2. General remarks. As already hinted at in the previous section, all finite injury or Π_1-constructions share the following features:

The *requirements* are typically of the form

$$(\neg P \to \neg R) \wedge (P \to S)$$

where P, R, and S are Σ_1-conditions, possibly relative to some computably enumerable oracles which we are building. (In some "nested" arguments, there may be more than two possibilities for the hypotheses, e.g., $\neg P$, $P \wedge \neg Q$, and $P \wedge Q$.

Correspondingly, these imply more conditions. We will restrict ourselves here to unnested arguments, i.e., arguments in which there is only one clause P. The last theorem in this section, the construction of a properly d-computably enumerable degree in Section 1.4, presents a nested argument.)

The *strategies* are of the following form:

1. Pick some parameters and start by ensuring $\neg R$.
2. Wait for P to appear true at some stage.
3. Now ensure S and protect the validity of P from now on.

The possible *outcomes* are then $\neg P$ and P. At each stage s, there is a *current outcome* (our current guess about P). The *true outcome* (i.e., the final outcome) is the current outcome at cofinitely many stages and, once current, remains current from then on. Let Λ be the (finite) set of possible outcomes of all strategies. (In the example of the Friedberg-Muchnik argument of the previous section, P is "$\exists s\,(\Phi_s(A_{1-i}; x) = 0)$", and R and S are $\exists s\,(x \in A_{i,s})$.

The *tree of strategies* is a finite branching subtree of $\Lambda^{<\omega}$ such that each node ("strategy") $\alpha \in T$ has as immediate successors all nodes $\alpha\hat{\ }\langle o\rangle$ where o ranges over the possible outcomes of α in Λ. The requirements are assigned effectively to the strategies in such a way that along any path through T, all requirements are handled. (Typically, each requirement is assigned to all strategies of a fixed length.)

The *construction* is performed in ω many stages s. Each stage consists of substages $t \leq s$. At substage t, the strategy α of length t eligible to act is chosen such that for all $\beta \subset \alpha$, $\beta\hat{\ }\langle o\rangle \subseteq \alpha$ iff β currently has current outcome o. Strategy α will then proceed according to its description from where it left off the last time it was eligible to act. We define the *current true path* f_s at stage s to be the longest strategy eligible to act at stage s.

For the *verification*, define the *true path* f to be the limit of the current true path. (This limit will exist since the true outcome of any strategy is current cofinitely often.) To verify the satisfaction of a requirement $\mathcal{R}$, consider the $\mathcal{R}$-strategy $\alpha \subset f$. α will first be eligible to act at some stage s_0, say, and pick its parameters at that stage. From then on, α will be eligible to act at each stage. Since all strategies $\beta \subset \alpha$ have achieved their final outcome by stage s_0, they will not injure α from then on. No β incomparable with α is eligible to act after stage s_0. And each $\beta \supset \alpha$ can only act when it currently has a correct guess about the current outcome of α and so will not injure α.

The combinatorial facts about Π_1-arguments and strategies are summarized in the following

Π_1-Lemma.

(i) *The true outcome of a Π_1-strategy is the current outcome at cofinitely many stages, and the true outcome, once current, must be current from then on.*

(ii) *If $\alpha \subseteq f_{s_0}$ (for some stage s_0) and $\alpha \subset f$ in a Π_1-construction then $\alpha \subseteq f_s$ for all stages $s \geq s_0$.*

Proof. (i) Clear from the description of Π_1-strategies above. (This will also be true for more complicated Π_1-strategies as in the construction of a properly d-computably enumerable degree.)

(ii) If $\beta \subset \alpha$ changes current outcome after stage s_0, then by (i), α can never be on the current true path again.

1.3. A low noncomputable computably enumerable degree. The following solution to Post's Problem is probably the easiest finite injury argument.

Theorem 2. *There is a low noncomputable computably enumerable degree* $\mathbf{a}$*, i.e., such that* $\mathbf{a} > \mathbf{0}$ *and* $\mathbf{a}' = \mathbf{0}'$*.*

Proof. We need to construct a computably enumerable set A and a computable function Γ meeting, for all partial computable functions Φ and all $e \in \omega$, the following

Requirements:

$$\mathcal{P}_\Phi : A \neq \Phi, \text{ and}$$
$$\mathcal{N}_e : A'(e) = \lim_t \Gamma(e,t).$$

By Shoenfield's Limit Lemma, the $\mathcal{N}$-requirements will ensure lowness.

Strategy for $\mathcal{P}_\Phi$:

1. Pick an unused witness x (targeted for A) larger than any number mentioned so far in the construction and keep x out of A.
2. Wait for $\Phi(x) = 0$.
3. Enumerate x into A and stop.

Outcomes of the $\mathcal{P}$-strategy:

w: Wait at Step 2 forever: Then $A(x) = 0 \neq \Phi(x)$.
s: Stop at Step 3: Then $A(x) = 1 \neq 0 = \Phi(x)$.

Strategy for $\mathcal{N}_e$:

1. Start by setting $\Gamma(e,t) = 0$ at each stage t.
2. Wait for $\{e\}(A;e)$ to converge.
3. Protect $\{e\}(A;e)$ by restraining $A\restriction(u(A;e,e)+1)$ and set $\Gamma(e,t) = 1$ at each stage t from now on (where $u(A;e,e)$ is the use of $\{e\}(A;e)$).

Outcomes of the $\mathcal{N}$-strategy:

w: Wait at Step 2 forever: Then $A'(e) = 0$ and $\Gamma(e,t) = 0$ for all t.
s: Reach Step 3: Then $A'(e) = 1$ and $\Gamma(e,t) = 1$ for cofinitely many t.

We let $\Lambda = \{s, w\}$ be the set of outcomes.

Tree of strategies: Let $T = \Lambda^{<\omega}$ be the tree of strategies. Effectively order the requirements (of order type ω) and assign to all strategies $\alpha \in T$ of length e the eth requirement in this list.

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$, and let α act according to the above description.

Verification:

Lemma. *Every strategy* $\alpha \subset f$ *ensures the satisfaction of its requirement.*

Proof. Fix s_0 least such that $\alpha \subseteq f_{s_0}$ (and so $\alpha \subseteq f_s$ for all $s \geq s_0$ by the Π_1-Lemma).

We distinguish cases by the type of requirement α works on:

Case 1: α is a $\mathcal{P}_\Phi$-strategy: Then α picks a witness x at stage s_0. If α always waits at Step 2 then $A(x) = 0 \neq \Phi(x)$. Otherwise, α eventually stops at Step 3, and so $A(x) = 1 \neq 0 = \Phi(x)$.

Case 2: α is an $\mathcal{N}_e$-strategy: Then α alone defines $\Gamma(e,t)$ (for any t) after stage s_0. If α always waits at Step 2 then $A'(e) = 0$ and $\Gamma(e,t) = 0$ for all $t \geq s_0$. Otherwise, α stops at Step 3 at some stage $s_1 \geq s_0$, say. Then $\Gamma(e,t) = 1$ for all $t \geq s_1$, and $A'(e) = 1$ since no other strategy can injure the computation $\{e\}(A;e)$ by the usual argument.

1.4. A properly d-computably enumerable degree.

A *d-computably enumerable* degree is the Turing degree of a difference of two computably enumerable sets, i.e., of a set $A = A_0 - A_1$ where A_0 and A_1 are computably enumerable.

Theorem 3. (Cooper (1971)) *There is a properly d-computably enumerable degree, i.e., a d-computably enumerable degree which is not computably enumerable.*

Proof. We need to construct a d-computably enumerable set A, meeting for all computably enumerable sets W and all Turing functionals Φ and Ψ the following

Requirements:

$$\mathcal{R}_{W,\Phi,\Psi} : A \neq \Phi^W \vee W \neq \Psi^A.$$

Strategy for $\mathcal{R}_{W,\Phi,\Psi}$:

1. Pick an unused witness x (targeted for A) larger than any number mentioned so far in the construction and keep x out of A.
2. Wait for
$$A(x) = \Phi^W(x) \wedge W\restriction(\varphi(x)+1) = \Psi^A\restriction(\varphi(x)+1).$$
3. Enumerate x into A and restrain $A\restriction(\psi\varphi(x)+1)$ (except for x entering A).
4. Wait again for
$$A(x) = \Phi^W(x) \wedge W\restriction(\varphi(x)+1) = \Psi^A\restriction(\varphi(x)+1).$$
5. Extract x from A, protect the restored computations $\Psi^A\restriction(\varphi(x)+1)$ by restraining $A\restriction(\psi\varphi(x)+1)$ (except for x leaving A), and stop.

(The conditions P and Q mentioned in the general remarks in Section 1.2 are the conditions mentioned in Steps 2 and 4 above, respectively.)

Outcomes of the $\mathcal{R}$-strategy:

w_0: Wait at Step 2 forever: Then $A(x) = 0 \neq \Phi^W(x)$ or $W\restriction(\varphi(x)+1) \neq \Psi^A\restriction(\varphi(x)+1)$.

w_1: Wait at Step 4 forever: Then $A(x) = 1 \neq \Phi^W(x)$ or $W\restriction(\varphi(x)+1) \neq \Psi^A\restriction(\varphi(x)+1)$.

s: Stop at Step 5: Then
$$W\restriction(\varphi(x)+1) \neq W\restriction(\varphi(x)+1)[s_1] = \Psi^A\restriction(\varphi(x)+1)[s_1] = \Psi^A\restriction(\varphi(x)+1)$$
where s_1 is the stage at which Step 3 was reached.

We let $\Lambda = \{s, w_1, w_0\}$ be the set of outcomes. Note that the Π_1-Lemma still applies to this construction.

Tree of strategies: Let $T = \Lambda^{<\omega}$ be the tree of strategies. Effectively order the requirements (of order type ω) and assign to all strategies $\alpha \in T$ of length e the eth requirement in this list.

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$, and let α act according to the above description.

Verification:

Lemma. *Every strategy $\alpha \subset f$ ensures the satisfaction of its requirement.*

Proof. Fix s_0 least such that $\alpha \subseteq f_{s_0}$ (and so $\alpha \subseteq f_s$ for all $s \geq s_0$ by the Π_1-Lemma).

Then α picks a witness at stage s_0 for which it proceeds according to the above strategy. Under true outcome w_0 or w_1, the satisfaction of the requirements is clear. Under outcome s, we observe that, by the usual argument, the computations $\Psi^A \restriction (\varphi(x) + 1)$ cannot be destroyed by the action of other strategies.

2. Degenerate Infinite Injury or Σ_2-Constructions

2.1. General remarks. Degenerate infinite injury or Σ_2-constructions share some of the properties of finite injury constructions as well as some of the properties of infinite injury constructions: On the one hand, each strategy is finitary (unless a hypothesis of the theorem turns out to be wrong, e.g., a set turns out to be computable that was assumed to be noncomputable). On the other hand, the conditions in the requirements are Π_2-conditions as in infinite injury constructions.

All degenerate infinite injury or Σ_2-constructions share the following features:

The *requirements* are typically of the form

$$(P \to S)$$

where $P \equiv \forall n P_0(n)$ and $S \equiv \forall n S_0(n)$ are Π_2-conditions, possibly relative to some computably enumerable oracles which we are building, and where the Π_2-condition S contradicts the hypotheses of the theorem so that if P is found true then we contradict the theorem's hypotheses and so the remaining requirements need not be satisfied.

A Σ_2-*strategy* is of the form

1. Pick some parameters and set $n = 0$.
2. Pick some more parameters (depending on n) and wait for $P_0(n)$.
3. Ensure another instance of S, preserve the validity of $P_0(n)$, increment n by $+1$, and go back to Step 2.

The possible *outcomes* are then the *finitary outcomes* $\neg P_0(n)$ (i.e., eventually waiting at Step 2 forever for some fixed n) and the *infinitary outcome* P (i.e., going from Step 3 to Step 2 infinitely often). At each stage s, there is a current outcome (the current guess that $\neg P_0(n)$). A true (i.e., final) finitary outcome is the current outcome at cofinitely many stages and, once current, remains current from then on. (The infinitary outcome does not have that property but it contradicts the theorem's hypotheses anyhow.) The set of *finitary* outcomes of a Σ_2-strategy will typically be an infinite set of order type ω^*, the set of natural numbers under the reverse ordering. (Strictly speaking, we will not need an ordering of outcomes at this point. However, the ordering aids the intuition in that the current outcome of Π_1- and Σ_2-strategies never moves "right" .) Let Λ be the set of possible *finitary* outcomes of all strategies (possibly including the outcomes of other Π_1-strategies).

The *tree of strategies* is an infinite branching subtree of $\Lambda^{<\omega}$ such that each node ("strategy") $\alpha \in T$ has as immediate successors all nodes $\alpha\hat{\ }\langle o \rangle$ where o ranges over the possible outcomes of α. The requirements are assigned effectively to the strategies in such a way that along any path through T, all requirements are handled. (Note that we do not put the infinitary outcome of Σ_2-strategies on the tree of strategies since under this outcome, the theorem's hypotheses are violated, and so we need not deal with further requirements.)

Again, the *construction* is performed in ω many stages s. Each stage consists of substages $t \leq s$. At substage t, the strategy α of length t eligible to act is chosen such that for all $\beta \subset \alpha$, $\beta\hat{}\langle o\rangle \subseteq \alpha$ iff β currently has current outcome o. Strategy α will then proceed according to its strategy from where it left off the last time it was eligible to act. We define the *current true path* f_s at stage s to be the longest strategy eligible to act at stage s.

For the *verification*, define the *true path* f to be the limit of the current true paths. (This limit will exist only if no strategy along it has the infinitary outcome. In the latter case, we let the true path be a node of the tree, namely, the longest node such that we can define this limit; so this node will have infinitary outcome.) To inductively verify the satisfaction of a requirement $\mathcal{R}$, consider the $\mathcal{R}$-strategy $\alpha \subseteq f$. Strategy α will first be eligible to act at some stage s_0, say, and pick its parameters at that stage. From then on, α will be eligible to act at each stage. Since all strategies $\beta \subset \alpha$ have achieved their final outcome by stage s_0, they will not injure α from then on. No β incomparable with α is eligible to act after stage s_0. And each $\beta \supset \alpha$ can only act when it currently has a correct guess about the current outcome of α and so will not injure α. Thus α will either eventually settle on a finitary outcome, allowing f to have length greater than $|\alpha|$; or α will have infinitary outcome, contradicting the theorem's hypotheses.

The combinatorial facts about Σ_2-arguments and strategies are summarized in the following

Σ_2-Lemma.

(i) *A true finitary outcome of a Σ_2-strategy is the current outcome at cofinitely many stages, and this outcome, once current, must be current from then on.*

(ii) *If $\alpha \subseteq f_{s_0}$ (for some stage s_0) and $\alpha \subseteq f$ in a Σ_2-construction then $\alpha \subseteq f_s$ for all stages $s \geq s_0$.*

Proof. (i) Clear from the description of Σ_2-strategies above.

(ii) If $\beta \subset \alpha$ changes current outcome after stage s_0, then by (i), α can never be on the current true path again.

2.2. Avoiding upper cones. Probably the easiest Σ_2-constructions are obtained by building computably enumerable degrees avoiding upper cones of computably enumerable degrees (above a fixed noncomputable computably enumerable degree). We give the easiest such example employing what is known as the *Sacks preservation strategy.*

Theorem 4. (Sacks) *For any noncomputable computably enumerable degree* $\mathbf{w}$*, there is a noncomputable computably enumerable Turing degree* $\mathbf{a} \not\geq \mathbf{w}$*.*

Proof. Fix a computably enumerable set W of degree $\mathbf{w}$. We need to construct a computably enumerable set A, meeting, for all partial computable functions Φ and all Turing functionals Ψ, the following

Requirements:

$$\mathcal{P}_\Phi : A \neq \Phi, \text{ and}$$
$$\mathcal{N}_\Psi : W = \Psi(A) \rightarrow \exists \Delta (W = \Delta).$$

Here Δ is a partial computable function built by us (i.e., by one of our strategies).

Strategy for $\mathcal{P}_\Phi$: This is the Π_1-strategy we have already encountered in the construction of a low noncomputable computably enumerable degree.

1. Pick an unused witness x (targeted for A) larger than any number mentioned so far in the construction and keep x out of A.
2. Wait for $\Phi(x) = 0$.
3. Enumerate x into A and stop.

Outcomes of the $\mathcal{P}$-strategy:

w: Wait at Step 2 forever: Then $A(x) = 0 \neq \Phi(x)$.
s: Stop at Step 3: Then $A(x) = 1 \neq 0 = \Phi(x)$.

Strategy for $\mathcal{N}_\Psi$: The strategy builds a "local" partial computable function (i.e., only this strategy may make definitions for Δ whereas other $\mathcal{N}$-strategies define their own versions of Δ).

1. Set $n = 0$.
2. Wait for $\Psi(A)\restriction(n+1) = W\restriction(n+1)$.
3. Preserve $\Psi(A)\restriction(n+1)$ by restraining numbers from entering $A\restriction(\psi(n)+1)$, set $\Delta(n) = W(n)$, increment n by $+1$, and go back to Step 2.

Outcomes of the $\mathcal{N}$-strategy:

w_n: Wait at Step 2 forever for some n: Then $W\restriction(n+1) \neq \Psi(A)\restriction(n+1)$.
∞: Eventually reach Step 3 for each n: Then $W = \Psi(A) = \Delta$, and so W is computable contrary to hypothesis.

We let $\Lambda = \{\dots, w_1, w_0\} \cup \{s, w\}$ be the set of finitary outcomes.

Tree of strategies: Effectively order the requirements (of order type ω). Inductively define a tree $T \subseteq \Lambda^{<\omega}$ by assigning to all strategies α of length e the eth requirement in this list and by letting $\alpha\hat{\ }\langle o\rangle$ be the immediate successors of α where o ranges over all possible finitary outcomes of α.

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$.

Verification: Denote by $f \in [T] \cup T$ the true path of the construction, i.e., the limit of f_s as s tends to infinity. Recall that f may be of finite length, i.e., be a node on the tree with infinitary outcome. We then only need to verify the satisfaction of the requirements handled by strategies $\subseteq f$.

Lemma. *Every strategy $\alpha \subseteq f$ ensures the satisfaction of its requirement, and if $\alpha = f$ then α contradicts the theorem's hypothesis.*

Proof. Fix s_0 least such that $\alpha \subseteq f_{s_0}$ (and so $\alpha \subseteq f_s$ for all $s \geq s_0$ by the Σ_2-Lemma).

We now distinguish cases for α.

Case 1: α is a $\mathcal{P}$-strategy: Then α picks a witness x at stage s_0, and it enumerates x into A iff $\Phi(x) = 0$.

Case 2: α is an $\mathcal{N}$-strategy: Then α starts defining its function Δ at stage s_0. If α eventually stops at Step 2 for some n then clearly $W \neq \Psi(A)$. Otherwise, α will make Δ total. Δ must correctly compute W since as in the proof of the original Friedberg-Muchnik Theorem, we can argue that a computation $\Psi(A; n)$ cannot be destroyed once found, and that any later $W(n)$-change must cause α to stop at Step 2 for some $n' > n$.

2.3. The Friedberg-Muchnik Theorem with permitting. Another easy technique for Σ_2-constructions consists in performing a finite injury argument below a fixed noncomputable computably enumerable degree. (This technique is known as *Friedberg permitting.*) We will illustrate this with the example of the Friedberg-Muchnik Theorem.

Theorem 5. *For any noncomputable computably enumerable degree* $\mathbf{w}$*, there are two incomparable computably enumerable Turing degrees* $\mathbf{a}_0, \mathbf{a}_1 \leq \mathbf{w}$*. (Thus no initial segment of the computably enumerable degrees is linearly ordered, and in particular the computably enumerable degrees are downward dense.)*

Proof. Fix a computably enumerable set W of degree $\mathbf{w}$. We need to construct two computably enumerable sets A_0 and A_1 as well as Turing functionals Γ_0 and Γ_1, meeting, for all $i \leq 1$ and all Turing functionals Φ, the following

Requirements:

$$\mathcal{R}^i : A_i = \Gamma_i(W), \text{ and}$$
$$\mathcal{P}^i_\Phi : A_i = \Phi(A_{1-i}) \to \exists \Delta (W = \Delta).$$

Here Δ is a partial computable function built by us (i.e., by one of our strategies).

Strategy for $\mathcal{R}^i$: $\mathcal{R}^i$ is a global requirement. Its strategy declares, at the end of each stage s and for each $i \leq 1$, a new computation $\Gamma_i(W; s) = A_i(s)$ with use $\gamma_i(s) = s$, and redefines $\Gamma_i(W; x) = A_i(x)$ with the previous use $\gamma_i(x) = x$ for arguments $x < s$ for which $\Gamma_i(W)$ has become undefined during stage s due to a W-change. This global strategy will not be put on the tree and has no outcomes.

Strategy for $\mathcal{P}^i_\Phi$: The strategy builds a "local" partial computable function Δ (i.e., only this strategy may make definitions for Δ whereas other $\mathcal{P}$-strategies define their own versions of Δ).

1. Set $n = 0$.
2. Pick an unused witness x_n (targeted for A_i) larger than any number mentioned so far in the construction and keep x_n out of A_i.
3. Wait for $\Phi(A_{1-i}; x_n) = 0$.
4. Preserve $\Phi(A_{1-i}; x_n) = 0$ by restraining numbers from $A_{1-i}\restriction(\varphi(x_n) + 1)$, set $\Delta(y) = W(y)$ for all arguments $y \in (\gamma_i(x_{n-1}), \gamma_i(x_n)]$ (where $\gamma_i(x_{-1}) = -1$), and go back to Step 2 with $n+1$ while simultaneously waiting at Step 5 with n.
5. Wait for W to *permit* x_n, i.e., for $W\restriction(\gamma_i(x_n) + 1)$ to change.
6. Enumerate x_n into A_i and stop the strategy (including the action for any $n' \neq n$).

Outcomes:

w_n: Wait at Step 3 forever for some n: Then $A_i(x_n) = 0 \neq \Phi(A_{1-i}; x_n)$.
s: Stop at Step 6 (for some n): Then $A_i(x_n) = 1 \neq 0 = \Phi(A_{1-i}; x_n)$.
∞: Eventually wait at Step 5 for each n: Then $W = \Delta$, and so W is computable contrary to hypothesis.

The current outcome of a $\mathcal{P}^i_\Phi$-strategy is s if the strategy has already stopped, and w_n otherwise where n is the current value of the parameter n.

We let $\Lambda = \{s, \dots, w_1, w_0\}$ be the set of finitary outcomes.

Tree of strategies: Let $T = \Lambda^{<\omega}$ be the tree of strategies. Effectively order the $\mathcal{P}$-requirements (of order type ω) and assign to all strategies $\alpha \in T$ of length e the eth requirement in this list.

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$, and let α act according to the description above. At the end of each stage s, the "global" strategies for the requirements $\mathcal{R}^0$ and $\mathcal{R}^1$ act as described above.

Verification: We first note that the $\mathcal{R}$-requirements are satisfied.

Lemma 1. *For each $i \leq 1$, $\Gamma_i(W) = A_i$.*

Proof. A number x is enumerated into a set A_i only at a stage at which $W \upharpoonright (\gamma_i(x) + 1)$ changes. So $\Gamma_i(W; x)$ can be redefined correctly at the end of each stage. Since the use $\gamma_i(x) = x$ is fixed, and since $\Gamma_i(W; x)$ is redefined at the end of each stage $\geq s$ if undefined, $\Gamma_i(W; x)$ is also defined for all x.

Denote by $f \in [T] \cup T$ the true path of the construction, i.e., the limit of f_s as s tends to infinity. Recall that f may be of finite length, i.e., be a node on the tree with infinitary outcome. We then only need to verify the satisfaction of the requirements handled by strategies $\subseteq f$.

Lemma 2. *Every strategy $\alpha \subseteq f$ ensures the satisfaction of its requirement, and if $\alpha = f$ then α contradicts the theorem's hypothesis.*

Proof. Fix s_0 least such that $\alpha \subseteq f_{s_0}$ (and so $\alpha \subseteq f_s$ for all $s \geq s_0$ by the Σ_2-Lemma).

Then α picks a witness x_0 at stage s_0.

Case 1: α eventually waits at Step 3 with some fixed n: Then $A_i(x_n) = 0 \neq \Phi(A_{1-i}; x_n)$.

Case 2: α eventually stops at Step 6 (with some n): Then $A_i(x_n) = 1 \neq 0 = \Phi(A_{1-i}; x_n)$ since as in the proof of the original Friedberg-Muchnik Theorem, we can argue that the computation $\Phi(A_{1-i}; x_n) = 0$ cannot be destroyed once found.

Case 3: α eventually waits at Step 5 for each n: Then clearly $\Delta = W$ (contradicting the noncomputability of W) since any $W(y)$-change once $\Delta(y)$ is defined would allow us to proceed to Step 6 for some n.

2.4. The Sacks Splitting Theorem.

The Sacks Splitting Theorem gives at the same time a set-theoretical and a degree-theoretical splitting of a noncomputable computably enumerable set. (Note that for disjoint computably enumerable sets A_0 and A_1, $\deg A_0 \cup \deg A_1 = \deg(A_0 \cup A_1)$.)

The construction for this theorem differs from the previous arguments given in that the priority ordering of the requirements is not arbitrary and must differ for different paths through the priority tree. (This is because here we cannot choose the numbers targeted for A_0 and A_1 arbitrarily but must use the numbers entering A.)

Theorem 6. (Sacks (1963)) *For any computably enumerable sets $V >_T \emptyset$ and W, there are computably enumerable sets $A_0, A_1 \not\geq_T V$ such that W is the disjoint union of A_0 and A_1. (Thus, setting $V = W$, any noncomputable computably enumerable degree $\mathbf{w}$ is the join of two incomparable computably enumerable degrees $\mathbf{a}_0, \mathbf{a}_1 < \mathbf{w}$.)*

Proof. We need to construct two computably enumerable set A_0 and A_1, meeting, for all $x \in \omega$ and all Turing functionals Φ, the following

Requirements:

$$\mathcal{R}: A_0, A_1 \subseteq W \wedge A_0 \cap A_1 = \emptyset,$$
$$\mathcal{P}_x: x \in W \to x \in A_0 \cup A_1, \text{ and}$$
$$\mathcal{N}^i_\Phi: V = \Phi(A_i) \to \exists \Delta (V = \Delta).$$

Here Δ is a partial computable function built by us (i.e., by one of our strategies).

Strategy for $\mathcal{R}$: This is a trivial global requirement which only allows numbers already in W to enter at most one of A_0 and A_1.

Strategy for $\mathcal{P}_x$: This is a trivial Π_1-strategy.

1. Wait for x to enter W.
2. Ensure that x is enumerated into one of A_0 and A_1 and stop.

Outcomes of the $\mathcal{P}_x$-strategy:

w: Wait at Step 1 forever: Then $W(x) = 0 = (A_0 \cup A_1)(x)$.
s: Stop at Step 2: Then $W(x) = 1 = (A_0 \cup A_1)(x)$.

Strategy for $\mathcal{N}^i_\Phi$: The strategy builds a "local" partial computable function (i.e., only this strategy may make definitions for Δ whereas other $\mathcal{N}$-strategies define their own versions of Δ).

1. Set $n = 0$.
2. Wait for $\Phi(A_i)\restriction(n+1) = V\restriction(n+1)$.
3. Preserve $\Phi(A_i)\restriction(n+1)$ by restraining numbers from entering $A_i\restriction(\varphi(A_i; n)+1)$, set $\Delta(n) = V(n)$, increment n by $+1$, and go back to Step 2.

Outcomes of the $\mathcal{N}$-strategy:

w_n: Wait at Step 2 forever for some n: Then $V\restriction(n+1) \neq \Phi(A_i)\restriction(n+1)$.
∞: Eventually reach Step 3 for each n: Then $V = \Phi(A_i) = \Delta$, and so V is computable contrary to hypothesis.

In this argument, however, we will need to use outcomes for the $\mathcal{N}$-strategy that contain more information; specifically, we will use the current overall A_i-restraint of the $\mathcal{N}$-strategy as its current outcome at a stage. This is because the assignment of strategies on the tree is no longer arbitrary as in previous constructions but depends on the restraint of higher-priority $\mathcal{N}$-strategies. So we let $\Lambda = \omega^* \cup \{s, w\}$ be the set of finitary outcomes where ω^* is the set of natural numbers under the reverse ordering.

Tree of strategies: As hinted at above, for this argument, we need to use a dynamic priority ordering of the requirements, i.e., one that depends on the path through the tree of strategies taken. This is because the $\mathcal{P}_x$-strategies cannot choose the number they will enumerate large as usual but must enumerate their fixed number x.

So fix an arbitrary effective priority ordering of the $\mathcal{N}$-requirements only. We inductively define a tree $T \subseteq \Lambda^{<\omega}$ and the assignment of requirements to all strategies $\alpha \in T$. We start by assigning to the empty node $\emptyset \in T$ the highest-priority $\mathcal{N}$-requirement and letting its immediate successors be $\langle r \rangle$ for $r \in \omega^*$. Assume that we have determined that $\alpha \in T$ and have assigned a requirement to $\alpha^- = \alpha\restriction(|\alpha| - 1)$. If α^- is assigned an $\mathcal{N}$-requirement then we assign to α the requirement $\mathcal{P}_x$ for the least x such that $\mathcal{P}_x$ is not assigned to any $\beta \subset \alpha$. Otherwise, let r be the greatest integer such that $\beta\hat{}\langle r \rangle \subseteq \alpha$ for some $\beta \subset \alpha$. If there is some (least) $x \leq r$ such that $\mathcal{P}_x$ is not assigned to any $\beta \subset \alpha$ then we assign $\mathcal{P}_x$ to α, else we assign to α the highest-priority $\mathcal{N}$-requirement not assigned to any $\beta \subset \alpha$.

We can now formulate a lemma about the assignment of requirements:

Lemma 1.

(i) *For any path $p \in [T]$ and any $\mathcal{P}$- or $\mathcal{N}$-requirement, there is a strategy $\alpha \subset p$ assigned to that requirement.*

(ii) *For any strategy $\alpha \in T$ assigned to a requirement $\mathcal{P}_x$, say, there is $i_\alpha \leq 1$ such that there is no $\mathcal{N}^{i_\alpha}$-strategy $\beta \subset \alpha$ with $\beta\hat{\ }\langle r\rangle \subseteq \alpha$ for some $r \geq x$.*

Proof. (i) An easy induction argument.

(ii) For the sake of a contradiction, assume, for each $i \leq 1$, there is an $\mathcal{N}^i$-strategy $\beta_i \subset \alpha$ with $\beta_i\hat{\ }\langle r_i\rangle \subseteq \alpha$ for some $r_i \geq x$. Fix i such that $\beta_i \subset \beta_{1-i}$. But then, by our assignment of requirements to strategies above, $\mathcal{P}_x$ would have been assigned before the $\mathcal{N}^{1-i}$-requirement.

We note that for some $\mathcal{P}$-strategies α, i_α may be chosen equal either 0 or 1. From now on, fix an effective assignment of indices i_α to all $\mathcal{P}$-strategies $\alpha \in T$ as in Lemma 1(ii).

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$, and let α act according to its description above. A $\mathcal{P}_x$-strategy will enumerate x into A_{i_α} when reaching Step 2 unless x is already in $A_0 \cup A_1$. The current outcome of an $\mathcal{N}^i$-strategy will be its current A_i-restraint.

Verification: Clearly, requirement $\mathcal{R}$ is met.

Denote by $f \in [T] \cup T$ the true path of the construction, i.e., the limit of f_s as s tends to infinity. Recall that f may be of finite length, i.e., be a node on the tree with infinitary outcome. We then only need to verify the satisfaction of the requirements handled by strategies $\subseteq f$.

Lemma 2. *Every strategy $\alpha \subseteq f$ ensures the satisfaction of its requirement, and if $\alpha = f$ then α contradicts the theorem's hypothesis.*

Proof. Fix s_0 least such that $\alpha \subseteq f_{s_0}$ (and so $\alpha \subseteq f_s$ for all $s \geq s_0$ by the Σ_2-Lemma).

We now distinguish cases for α.

Case 1: α is a $\mathcal{P}_x$-strategy: Then α enumerates x into $A_0 \cup A_1$ iff $x \in W$.

Case 2: α is an $\mathcal{N}^i_\Phi$-strategy: Then α starts defining its function Δ at stage s_0. Let r be the greatest integer such that $\beta\hat{\ }\langle r\rangle \subseteq \alpha$ for some $\beta \subset \alpha$, or -1 if no such β exists. Note that by the way we assigned requirements, no number $\leq r$ enters $A_0 \cup A_1$ after stage s_0.

If α eventually stops at Step 2 for some n then clearly $V \neq \Phi(A_i)$. Otherwise, α will make Δ total. If we can show that no computation $\Phi(A_i; n)$ can be destroyed once α defines $\Delta(n)$ then we can argue as in the theorem on avoiding upper cones that Δ must correctly compute V.

Suppose some $\mathcal{P}$-strategy β enumerates a number x into A_i, destroying a computation $\Phi(A_i; n)$ once α has defined $\Delta(n)$. Then $\beta \supseteq \alpha\hat{\ }\langle r\rangle$ for some $r \geq \varphi(A_i; n)$ (since no other strategy β can enumerate any number after the stage at which α defines $\Delta(n)$), and $x \leq \varphi(A_i; n)$, contradicting the definition of i_β.

3. Infinite Injury or Π_2-Constructions

3.1. General remarks. In infinite injury or Π_2-constructions, strategies may be infinitary, i.e., they may (i) enumerate an infinite set of numbers, or (ii) impose unbounded restraint. Lower-priority strategies will have to deal with this behavior, typically by (i) guessing the set of numbers enumerated by higher-priority strategies, and (ii) acting when the higher-priority strategies' restraint is at a minimum. For this, the lower-priority strategies will need a fairly detailed guess about the higher-priority strategies' behavior which is given by the tree structure.

A second difference between infinite injury constructions on the one hand, and finite or degenerate infinite injury constructions on the other hand, is that a strategy on the true path need no longer be on the current true path at cofinitely many stages but rather only at an infinite number of stages. (This is because the outcomes involve Π_2-conditions which cannot be effectively approximated by a single limit.)

We will introduce a total ordering of the tree by defining a linear ordering $<_\Lambda$ on the set of outcomes Λ, which then induces a lexicographical ordering of the tree of strategies T. We will denote by $\alpha <_L \beta$ that there are $\gamma \in T$ and $o_\alpha <_\Lambda o_\beta$ such that $\gamma\hat{\ }\langle o_\alpha\rangle \subseteq \alpha$ and $\gamma\hat{\ }\langle o_\beta\rangle \subseteq \beta$; by $\alpha < \beta$ the *priority ordering* that $\alpha <_L \beta$ or $\alpha \subset \beta$; and by $\alpha \leq \beta$ that $\alpha <_L \beta$ or $\alpha \subseteq \beta$.

We will now be able to define the true path as the "lim inf" of the current true path by the following inductive definition:

$$f(n) = \liminf_s \{o \in \Lambda \mid \exists^\infty s((f\restriction n)\hat{\ }\langle o\rangle \subseteq f_s)\}.$$

So if a strategy α is on the true path f then the strategies $\beta <_L \alpha$ collectively are eligible to act at only finitely many stages; α has correct guesses about the strategies $\beta \subset \alpha$; and α has higher priority than the strategies $\beta > \alpha$. (In fact, as we will see in the Π_2-Lemma below, once a node $\alpha \subset f$ first appears on the current true path, no $\beta <_L \alpha$ will be eligible to act any more.)

The *requirements* are typically of the form

$$(\neg P \to \neg R) \wedge (P \to S)$$

where $P \equiv \forall n P_0(n)$, $R \equiv \forall n R_0(n)$, and $S \equiv \forall n S_0(n)$ are Π_2-conditions, possibly relative to some computably enumerable oracles which we are building. (In contrast to degenerate infinite injury constructions, the Π_2-condition P may be true without contradicting any hypothesis of the theorem.)

A Π_2-*strategy* is of the form

1. Pick some parameters and set $n = 0$.
2. Pick some more parameters (depending on n) and wait for $P_0(n)$ while ensuring $\neg R$.
3. Protect the validity of $P_0(n)$ from now on, ensure another instance of S, increment n by $+1$, and go back to Step 2.

The possible *outcomes* are then the *finitary outcomes* $\neg P_0(n)$ (i.e., eventually waiting at Step 2 forever for some fixed n) and the *infinitary outcome* P (i.e., going from Step 3 to Step 2 infinitely often). At each stage s, there is a *current outcome* which is either the current guess $\neg P_0(n)$ if, at stage s, the strategy just continued waiting at Step 2 with this n, or the guess P if, at stage s, the strategy found yet another instance of P to be true and moved from Step 2 to Step 3 back to Step 2.

We order the set of outcomes $\Lambda = \{P <_\Lambda \ldots <_\Lambda \neg P_0(2) <_\Lambda \neg P_0(1) <_\Lambda \neg P_0(0)\}$ and observe that the true outcome is the leftmost current outcome that is achieved infinitely often, namely, either the infinitary outcome P achieved at infinitely many stages, or a finitary outcome $\neg P_0(n)$ achieved at cofinitely many stages. In the constructions below, we will generally denote the outcomes P and $\neg P_0(n)$ by ∞ and w_n, respectively.

The *tree of strategies* is an infinite branching subtree of $\Lambda^{<\omega}$ such that each node ("strategy") $\alpha \in T$ has as immediate successors all nodes $\alpha\hat{\ }\langle o\rangle$ where o ranges over the possible outcomes of α. The requirements are assigned effectively to the strategies in such a way that along any path through T, all requirements are handled.

Again, the *construction* is performed in ω many stages s. Each stage consists of substages $t \leq s$. At substage t, the strategy α of length t eligible to act is chosen such that for all $\beta \subset \alpha$, $\beta\hat{\ }\langle o\rangle \subseteq \alpha$ iff β currently has current outcome o. Strategy α will then proceed according to its strategy from where it left off the last time it was eligible to act. We define the *current true path* f_s at stage s to be the longest strategy eligible to act at stage s.

For the *verification*, define the *true path* f to be the lim inf of the current true path in the sense given above, which will always be an infinite path through T. To inductively verify the satisfaction of a requirement $\mathcal{R}$, consider the $\mathcal{R}$-strategy $\alpha \subset f$. Strategy α will be eligible to act at some stage s_0, say. From then on, α will be eligible to act at infinitely many stages. Since no strategy $\beta <_\Lambda \alpha$ will be eligible to act after stage s_0, since α has a correct guess about all strategies $\beta \subset \alpha$, and since α has higher priority than all strategies $\beta > \alpha$, α will not be injured after stage s_0.

The combinatorial facts about Π_2-arguments and strategies are summarized in the following lemma. (The claims of this lemma can always be arranged to be true for infinite injury constructions; however, this will sometimes be at the expense of having very complicated outcomes for the strategies; so we will not always try to arrange to fully ensure (ii) and (iii) of this lemma.)

Π_2-Lemma.

(i) *A true finitary outcome of a Π_2-strategy is the current outcome at cofinitely many stages, and this outcome, once current, must be current from then on.*

(ii) *A true infinitary outcome of a Π_2-strategy is the current outcome at infinitely many stages. In the case of a true infinitary outcome, any current finitary outcome, once no longer current, can never be current again.*

(iii) *If $\alpha \subseteq f_{s_0}$ (for some stage s_0) and $\alpha \subset f$ in a Π_2-construction then $\alpha \leq f_s$ for all stages $s \geq s_0$.*

Proof. (i), (ii) Clear from the description of Π_2-strategies above. (This will also be true for more complicated Π_2-strategies as in the construction of an embedding of the nondistributive lattice M_3.)

(iii) If β with $\beta\hat{\ }\langle o\rangle \subseteq \alpha$ (for a finitary outcome o) changes current outcome after stage s_0, then by (i), α can never be on the current true path again. On the other hand, if $\beta\hat{\ }\langle o\rangle \subseteq \alpha$ (for the infinitary outcome o) then this is trivial by our ordering of Λ.

3.2. A high incomplete computably enumerable degree. Probably the easiest Π_2-construction is the construction of a high incomplete computably enumerable degree. It uses a variation of the first infinite injury priority argument, Shoenfield's Thickness Lemma (1961). (We defer the proof of the full Thickness Lemma to the next chapter since it is a Σ_3-construction.)

Theorem 7. (Sacks (1963)) *There is a high incomplete computably enumerable Turing degree, i.e., a computably enumerable degree* $\mathbf{h} < \mathbf{0}'$ *with* $\mathbf{h}' = \mathbf{0}''$.

Proof. Instead of showing that $\mathbf{h} \not\geq \mathbf{0}'$ directly, we will construct an auxiliary computably enumerable degree $\mathbf{c}$ ("our version" of the complete computably enumerable degree) and show $\mathbf{c} \not\leq \mathbf{h}$ using Friedberg-Muchnik strategies instead of the more complicated Sacks preservation strategies.

For highness, we use Shoenfield's Limit Lemma and the Π_2-complete index set $\mathrm{Inf} = \{e \mid W_e \text{ is infinite}\}$.

We thus need to construct computably enumerable sets C and H as well as a Turing functional Γ, meeting, for all $e \in \omega$ and all Turing functionals Φ the following

Requirements:

$$\mathcal{P}_e : \mathrm{Inf}(e) = \lim_v \Gamma(H; e, v), \text{ and}$$
$$\mathcal{N}_\Phi : C \neq \Phi(H).$$

Strategy for $\mathcal{P}_e$: The strategy initially defines $\Gamma(H; e, v)$ for each v at stage v by setting $\Gamma(H; e, v) = 0$ with use $\gamma(e, v)$ bigger than any number mentioned thus far in the construction. At the end of any stage $> v$, the strategy then redefines $\Gamma(H; e, v)$ to its previous value with the same use if it is now undefined due to an H-change.

Furthermore, the strategy proceeds as follows to potentially correct $\Gamma(H)$:

1. Set $n = 0$.
2. Wait for a number $\geq n$ to appear in W_e.
3. For all $v \leq$ the current stage s, enumerate $\gamma(e, v)$ into H and reset $\Gamma(H; e, v) = 1$ with the same use, increment n by $+1$, and go back to Step 2.

Outcomes of the $\mathcal{P}_e$-strategy:

- w_{n_0}: The strategy resets $\Gamma(H; e, v) = 1$ only for $v \leq$ some fixed s, and the parameter n reaches a finite limit n_0. Then $n_0 - 1 = \max(W_e)$ (setting $\max(\emptyset) = -1$), so W_e is finite. (These are the finitary outcomes.)
- ∞: The strategy resets $\Gamma(H; e, v) = 1$ for all v: Then the set W_e contains arbitrarily large numbers and is thus infinite. (This is the infinitary outcome.)

In either case, we clearly have $\mathrm{Inf}(e) = \lim_v \Gamma(H; e, v)$ as required.

We let the *current outcome* of the $\mathcal{P}$-strategy at a stage s be w_{n_0} if the parameter n remains unchanged at n_0 during stage s, and ∞ otherwise. So, by the Π_2-Lemma, a true finitary outcome of a strategy is the current outcome of the strategy at cofinitely many stages whereas a true infinitary outcome of the strategy is the current outcome only at infinitely many stages.

Strategy for $\mathcal{N}_\Phi$: This is just the Friedberg-Muchnik strategy, i.e., a Π_1-strategy:

1. Pick an unused witness x (targeted for C) larger than any number mentioned so far in the construction and keep x out of C.
2. Wait for $\Phi(H; x) = 0$.
3. Enumerate x into C, preserve $\Phi(H; x) = 0$ by restraining numbers from entering $H \upharpoonright (\varphi(x) + 1)$, and stop.

Outcomes of the $\mathcal{N}$-strategy:

w: Wait at Step 2 forever: Then $C(x) = 0 \neq \Phi(H; x)$.

s: Stop at Step 3: Then $C(x) = 1 \neq 0 = \Phi(H; x)$.

We let $\Lambda = \{\infty <_\Lambda \ldots <_\Lambda w_2 <_\Lambda w_1 <_\Lambda w_0 <_\Lambda s <_\Lambda w\}$ be the set of outcomes.

Tree of strategies: Effectively order the requirements (of order type ω). Inductively define a tree $T \subseteq \Lambda^{<\omega}$ by assigning to all strategies α of length e the eth requirement in this list and by letting $\alpha\hat{\ }\langle o\rangle$ be the immediate successors of α where o ranges over all possible outcomes of α.

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$. The strategy α will then act according to the above description, with three important modifications in the case that α is a $\mathcal{P}_e$-strategy: Suppose α was first eligible to act at a stage $s_0 \leq s$. Then (i) α will start at stage s_0 with its parameter n set to s_0 (instead of 0 as above); (ii) α will not reset $\Gamma(H; e, v)$ for any $v < s_0$; and (iii) α will never reset $\Gamma(H; e, v)$ from 1 to 0. (These modifications are necessary to ensure that the the $\mathcal{P}_e$-strategies off the true path do not reset $\Gamma(H)$ too often, and that the $\mathcal{N}$-strategies can enforce their H-restraint.)

Verification: Denote by $f \in [T]$ the true path of the construction, i.e., the lim inf of f_s as s tends to infinity. Note that by the above remarks about current outcomes, f will be an infinite path through T.

Lemma. *Every strategy $\alpha \subset f$ ensures the satisfaction of its requirement.*

Proof. We proceed by induction on the length of α. Fix s_0 least such that $\alpha \subseteq f_{s_0}$. Then $\alpha \leq f_s$ for all $s \geq s_0$ by the Π_2-Lemma.

We now distinguish cases for α.

Case 1: α is an $\mathcal{N}_\Phi$-strategy: Then α picks a permanent witness x at stage s_0. We distinguish two subcases:

Subcase 1.1: α never enumerates x into C: Then $\Phi(H; x) \neq 0 = C(x)$.

Subcase 1.2: α enumerates x into C at a stage $s_1 \geq s_0$: Then no strategy can injure α's H-restraint after stage s_1 since no $\beta <_L \alpha$ is eligible to act after stage s_0, no β with $\beta\hat{\ }\langle o\rangle \subseteq \alpha$ for $o \neq \infty$ enumerates a number after stage s_0, no β with $\beta\hat{\ }\langle\infty\rangle \subseteq \alpha$ enumerates a number $\leq s_1$ after stage s_1 (since it will have enumerated all possible numbers $\leq s_1$ by stage s_1 already), and no β with $\alpha < \beta$ enumerates a number $\leq s_1$ after stage s_1 (since β is not eligible to act before stage s_1 by the Π_1-Lemma and the Π_2-Lemma). Thus after stage s_1, α ensures $C(x) \neq \Phi(H; x)$.

Case 2: α is a $\mathcal{P}_e$-strategy: If W_e is infinite then α will reset $\Gamma(H; e, s) = 1$ for all $s \geq s_0$. On the other hand, if W_e is finite, say, $n_0 - 1 = \max(W_e)$, then only $\mathcal{P}_e$-strategies eligible to act before stage n_0 can possibly reset $\Gamma(H; e, v) = 1$ for any v; and, by the Π_2-Lemma, the only strategy among these possibly eligible to act infinitely often is α itself; thus $\Gamma(H; e, v)$ will be reset to 1 for at most finitely many v. (Note that many definitions of $\Gamma(H; e, v)$ will actually be made by strategies $\neq \alpha$.)

3.3. The Sacks Jump Inversion Theorem. Another variation of the first infinite injury priority argument, Shoenfield's Thickness Lemma (1961), is the Sacks Jump Inversion Theorem.

Theorem 8. (Sacks (1963)) *Given any degree $\mathbf{s} \geq \mathbf{0}'$ computably enumerable in $\mathbf{0}'$, there is a computably enumerable degree $\mathbf{a}$ with $\mathbf{a}' = \mathbf{s}$.*

Proof. Fix a Σ_2-set $S \in \mathbf{s}$ and a computable function h such that $e \in S$ iff $W_{h(e)}$ is finite, using that the index set $\text{Fin} = \{x \mid W_x \text{ is finite}\}$ is Σ_2-complete.

We now need to construct a computably enumerable set A and two Turing functionals Γ and Δ, meeting, for all $e \in \omega$, the following

Requirements:

$$\mathcal{P}_e : S(e) = \lim_v \Gamma(A; e, v), \text{ and}$$
$$\mathcal{N}_e : A'(e) = \Delta(S \oplus \emptyset'; e).$$

By Shoenfield's Limit Lemma, this will ensure $A' \equiv_T S$ ($\equiv_T S \oplus \emptyset'$) as desired.

Strategy for $\mathcal{P}_e$: (This is the same strategy as in the previous section, but with the roles of 0 and 1 reversed.) The strategy initially defines $\Gamma(A; e, v)$ for each v at stage v by setting $\Gamma(A; e, v) = 1$ with use $\gamma(e, v)$ bigger than any number mentioned thus far in the construction. At the end of any stage $> v$, the strategy then redefines $\Gamma(A; e, v)$ to its previous value with the same use if it is now undefined due to an A-change.

Furthermore, the strategy proceeds as follows to potentially correct $\Gamma(A)$:

1. Set $n = 0$.
2. Wait for a number $\geq n$ to appear in $W_{h(e)}$.
3. For all $v \leq$ the current stage s, enumerate $\gamma(e, v)$ into A and reset $\Gamma(A; e, v) = 0$ with the same use, increment n by $+1$, and go back to Step 2.

Outcomes of the $\mathcal{P}_e$-strategy:

w_{n_0}: The strategy resets $\Gamma(A; e, v) = 0$ only for $v \leq$ some fixed s, and the parameter n reaches a finite limit n_0. Then $n_0 - 1 = \max(W_{h(e)})$ (setting $\max(\emptyset) = -1$), so $e \in S$. (These are the finitary outcomes.)

∞: The strategy resets $\Gamma(A; e, v) = 0$ for all v: Then the set $W_{h(e)}$ contains arbitrarily large numbers, and thus $e \notin S$. (This is the infinitary outcome.)

We let the *current outcome* of the $\mathcal{P}$-strategy at a stage s be w_{n_0} if the parameter n remains unchanged at n_0 during stage s, and ∞ otherwise. So, by the Π_2-Lemma, a true finitary outcome of a strategy is the current outcome of the strategy at cofinitely many stages whereas a true infinitary outcome of the strategy is the current outcome only at infinitely many stages.

Strategy for $\mathcal{N}_e$: We begin by setting $\Delta(S \oplus \emptyset'; e) = 0$ with "appropriate" use (to be defined later). When a computation $\Phi_e(A; e)$ appears then we set $\Delta(S \oplus \emptyset'; e) = 1$ with "appropriate" use and restrain $A \upharpoonright (\varphi_e(e) + 1)$ to protect the computation $\Phi_e(A; e)$. (We will show below that we can always redefine $\Delta(S \oplus \emptyset'; e)$ whenever needed.)

Outcomes of the $\mathcal{N}$-strategy:

w: The strategy never finds a computation $\Phi_e(A; e)$: Then $A'(e) = 0 = \Delta(S \oplus \emptyset'; e)$.

s: The strategy finds a computation $\Phi_e(A; e)$: Then we protect this computation $\Phi_e(A; e)$ and so $A'(e) = 1 = \Delta(S \oplus \emptyset'; e)$.

We let the *current outcome* of an $\mathcal{N}_e$-strategy be w if the strategy is still searching for a computation $\Phi_e(A; e)$, and s otherwise.

We now let $\Lambda = \{\infty <_\Lambda \ldots <_\Lambda w_2 <_\Lambda w_1 <_\Lambda w_0 <_\Lambda s <_\Lambda w\}$ be the set of outcomes.

Tree of strategies: Effectively order the requirements (of order type ω). Inductively define a tree $T \subseteq \Lambda^{<\omega}$ by assigning to all strategies α of length i the ith requirement in this list and by letting $\alpha\hat{\ }\langle o\rangle$ be the immediate successors of α where o ranges over all possible outcomes of α.

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$. The strategy α will then act according to the above description, with two important modifications

If α is a $\mathcal{P}_e$-strategy, say, α was first eligible to act at a stage $s_0 \leq s$, then (i) α will start at stage s_0 with its parameter n set to s_0 (instead of 0 as above); (ii) α will not reset $\Gamma(A; e, v)$ for any $v < s_0$; and (iii) α will never reset $\Gamma(A; e, v)$ from 0 to 1. (These modifications are necessary to ensure that the the $\mathcal{P}_e$-strategies off the true path do not reset $\Gamma(A)$ too often, and that the $\mathcal{N}$-strategies can enforce their A-restraint.)

If α is an $\mathcal{N}_e$-strategy α then it will set $\Delta(X \oplus Y; e)$ equal to its guess about $A'(e)$ for all oracles $X \oplus Y$ satisfying:

(i) $X(i) = 0$ for all $\mathcal{P}_i$-strategies β with $\beta\hat{\ }\langle\infty\rangle \subseteq \alpha$;
(ii) $X(i) = 1$ for all $\mathcal{P}_i$-strategies β with $\beta\hat{\ }\langle n\rangle \subseteq \alpha$ for some $n \in \omega$;
(iii) $Y(c) = 0$ where c is a code for the Σ_1-question "Is there a future stage at which the current true path is to the left of α?";
(iv) $Y(c') = 1$ for all such codes c' used in prior definitions of Δ by an $\mathcal{N}_e$-strategy β which has since then been to the right of the current true path;
(v) $Y(d) = 0$ where d is a code for the Σ_1-question "Is there a future stage at which α is eligible to act and has a different guess about $A'(e)$?"; and
(vi) $Y(d') = 1$ for all such codes d' used before by α with a different guess about $A'(e)$.

(Note that by Kleene's Fixed-Point Theorem, we may fix such indices c and d during the construction. Note that the single computation $\Delta(S \oplus \emptyset'; e)$ is being defined by many different $\mathcal{N}_e$-strategies; the codes used in the oracle above ensure that there will be no incompatible definitions as verified in detail below; i.e., there may be several definitions using the same oracle but then they will also give the same value for $\Delta(S \oplus \emptyset'; e)$.)

Verification: Denote by $f \in [T]$ the true path of the construction, i.e., the lim inf of f_s as s tends to infinity. Note that f will be an infinite path through T.

Lemma. *Every strategy $\alpha \subset f$ ensures the satisfaction of its requirement.*

Proof. We proceed by simultaneous induction on the length of α. Fix s_0 least such that $\alpha \subseteq f_{s_0}$. Then $\alpha \leq f_s$ for all $s \geq s_0$ by the Π_2-Lemma.

We now distinguish cases for α.

Case 1: α is an $\mathcal{N}_e$-strategy: We first show that α can always define $\Delta(S \oplus \emptyset'; e)$ according to its strategy. Suppose this fails due to a prior definition of $\Delta(S \oplus \emptyset'; e)$ to a different value by a strategy (necessarily an $\mathcal{N}_e$-strategy) β at a stage s'. If $\beta > \alpha$ then $\beta >_L \alpha$ and the current true path has thus moved to the left of β since stage s', so β's code c is one of the codes c' used by α. If $\beta = \alpha$ then α has changed its guess about $A'(e)$, and β's code d is now one of the codes d' used by α. Finally, if $\beta < \alpha$ then $\beta <_L \alpha$. Fix the node ξ at which β and α split. ξ cannot be an $\mathcal{N}$-strategy by the Π_1-Lemma since β's outcomes cannot "change to the right". So assume that ξ is a $\mathcal{P}_i$-strategy; by the Π_2-Lemma, $\xi\hat{\ }\langle\infty\rangle \subseteq \beta$ and $\xi\hat{\ }\langle w_n\rangle \subseteq \alpha$ for some $n \in \omega$, and so β and α use a different value of $S(i)$ in their oracle for Δ.

Now let s_1 be the least stage $\geq s_0$ at which α finds a computation $\Phi_e(A; e)$ (if such a stage exists), and let $s_1 = s_0$ otherwise. Then at stage s_1, β will define $\Delta(S\oplus$

$\emptyset'; e) = A'(e)$ with correct $S \oplus \emptyset'$-oracle (and thus define $\Delta(S \oplus \emptyset'; e)$ permanently). Finally, no strategy can injure α's A-restraint after stage s_1 since no $\beta <_L \alpha$ is eligible to act after stage s_0, no β with $\beta\hat{\ }\langle o \rangle \subseteq \alpha$ for $o \neq \infty$ enumerates a number after stage s_0, no β with $\beta\hat{\ }\langle \infty \rangle \subseteq \alpha$ enumerates a number $\leq s_1$ after stage s_1 (since it will have enumerated all possible numbers $\leq s_1$ by stage s_1 already), and no β with $\alpha < \beta$ enumerates a number $\leq s_1$ after stage s_1 (since it is not eligible to act before stage s_1 as α permanently changes outcome at stage s_1). Thus $A'(e) = \Delta(S \oplus \emptyset'; e)$ as desired.

Case 2: α is a $\mathcal{P}_e$-strategy: If $W_{h(e)}$ is infinite then α will reset $\Gamma(A; e, s) = 0$ for all $s \geq s_0$. On the other hand, if $W_{h(e)}$ is finite, say, $n_0 - 1 = \max(W_{h(e)})$, then only $\mathcal{P}_e$-strategies eligible to act before stage n_0 can possibly reset $\Gamma(A; e, v) = 0$ for any v; and, by the Π_2-Lemma, the only strategy among these possibly eligible to act infinitely often is α itself; thus $\Gamma(A; e, v)$ will be reset to 0 for at most finitely many v. (Note that many definitions of $\Gamma(A; e, v)$ will actually be made by strategies $\neq \alpha$.)

3.4. A minimal pair of computably enumerable degrees. The minimal pair theorem is the first infinite injury argument where the tree method is crucial. (Earlier arguments can be carried out using the so-called *true stages method*, which argues that at every W*-true stage*, i.e., at every stage s at which an element x enters a set W such that $W \restriction x = W_s \restriction x$, any restraint function depending on W is no greater than what it will be from now on.)

The first refutation to Shoenfield's Conjecture (1965) was the following

Theorem 9. (Lachlan (1966) and Yates (1966)) *There is a minimal pair of computably enumerable Turing degrees, i.e., a pair of computably enumerable degrees* $\mathbf{a}_0, \mathbf{a}_1 > \mathbf{0}$ *with* $\mathbf{a}_0 \cap \mathbf{a}_1 = \mathbf{0}$.

Proof. We construct two computably enumerable sets A_0 and A_1, meeting, for all Turing functionals Φ and Ψ, the following

Requirements:

$$\mathcal{P}^i_\Phi : A_i \neq \Phi, \text{ and}$$
$$\mathcal{N}_\Psi : \Psi(A_0) = \Psi(A_1) \text{ is total} \to \exists \Delta(\Delta = \Psi(A_0)).$$

Here Δ is a partial computable function built by us (i.e., by one of our strategies). (We use *Posner's Trick* as usual for these arguments, allowing the same functional for both A_0 and A_1. This works since A_0 and A_1 must be distinct by the requirements as stated above, and so two functionals Ψ_0 and Ψ_1 can be simulated by a single functional Ψ.)

Strategy for $\mathcal{P}^i_\Phi$: This is the Π_1-strategy we have already encountered in the construction of a low noncomputable computably enumerable degree.

1. Pick an unused witness x (targeted for A_i) larger than any number mentioned so far in the construction and keep x out of A_i.
2. Wait for $\Phi(x) = 0$.
3. Enumerate x into A_i and stop.

Outcomes of the $\mathcal{P}$-strategy:

w: Wait at Step 2 forever: Then $A_i(x) = 0 \neq \Phi(x)$.
s: Stop at Step 3: Then $A_i(x) = 1 \neq 0 = \Phi(x)$.

As for any Π_1-strategy, the true outcome of the $\mathcal{P}$-strategy is the current outcome at cofinitely many stages.

Strategy for $\mathcal{N}_\Psi$: The strategy builds a "local" partial computable function (i.e., only this strategy may make definitions for Δ whereas other $\mathcal{N}$-strategies define their own versions of Δ).

1. Set $n = 0$.
2. Wait for $\Psi(A_0)\restriction(n+1) = \Psi(A_1)\restriction(n+1)$.
3. Set $\Delta(n) = \Psi(A_0; n)$, and preserve either $\Psi(A_0; n)$ or $\Psi(A_1; n)$ from now on; i.e., when both $\Psi(A_0; n)$ and $\Psi(A_1; n)$ are defined and equal to $\Delta(n)$ then protect at least one of them, and when only one of them is defined then protect it, by restraining $A_i\restriction(\psi(A_i; n) + 1)$ for the appropriate i.
4. Increment n by $+1$ and go back to Step 2.

Outcomes of the $\mathcal{N}$-strategy:

w_n: Wait at Step 2 forever for this n: Then $\Psi(A_0)\restriction(n+1) \neq \Psi(A_1)\restriction(n+1)$. (These are the finitary outcomes.)

∞: Eventually reach Step 4 for each n: Then clearly Δ is total. If furthermore $\Psi(A_0) = \Psi(A_1)$ is total, then by the restraint, Δ must compute their common value correctly. (This is the infinitary outcome.)

We let the current outcome of the $\mathcal{N}$-strategy be ∞ if the strategy has just incremented its parameter n; otherwise the current outcome is w_n for the current value of n. Then a true finitary outcome of the $\mathcal{N}$-strategy is the current outcome at cofinitely many stages whereas a true infinitary outcome is the true outcome only at infinitely many stages.

We let $\Lambda = \{\infty <_\Lambda \ldots <_\Lambda w_2 <_\Lambda w_1 <_\Lambda w_0 <_\Lambda s <_\Lambda w\}$ be the set of outcomes.

Tree of strategies: Effectively order the requirements (of order type ω). Inductively define a tree $T \subseteq \Lambda^{<\omega}$ by assigning to all strategies α of length e the eth requirement in this list and by letting $\alpha\hat{\ }\langle o\rangle$ be the immediate successors of α where o ranges over all possible outcomes of α.

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$. (The restraint of the $\mathcal{N}$-strategies is automatic and need not be made explicitly.)

Verification: Denote by $f \in [T]$ the true path of the construction, i.e., the lim inf of f_s as s tends to infinity.

Lemma. *Every strategy $\alpha \subset f$ ensures the satisfaction of its requirement.*

Proof. Fix s_0 least such that $\alpha \subseteq f_{s_0}$. Then $\alpha \leq f_s$ for all $s \geq s_0$ by the Π_2-Lemma.

We now distinguish cases for α.

Case 1: α is a $\mathcal{P}^i_\Phi$-strategy: Then α picks a permanent witness x at stage s_0. Now α ensures its requirement as in the basic module above.

Case 2: α is an $\mathcal{N}_\Psi$-strategy: Assume that $\Psi(A_0) = \Psi(A_1)$ is total. Then clearly α has outcome ∞, and so Δ is total. It thus remains to verify the correctness of Δ. We will show this by establishing that once $\Delta(n)$ is defined, it equals at least one of $\Psi(A_0; n)$ and $\Psi(A_1; n)$ at any stage.

Let $s_n \geq s_0$ be the stage at which α first sets its counting parameter to n. We first note that at any stage $s \geq s_n$, only a $\mathcal{P}$-strategy $\beta \supseteq \alpha\hat{\ }\langle\infty\rangle$ can possibly destroy a computation $\Psi(A_i; m)[s_n]$ (for $m < n$) that has existed at stage s_n since all $\beta < \alpha$ no longer enumerate numbers, and all $\beta >_L \alpha\hat{\ }\langle\infty\rangle$ which act at or after stage s_n have witnesses that are too large. But at any stage, at most one number is enumerated into at most one set since once a $\mathcal{P}$-strategy β enumerates a number, only strategies $\beta' \supseteq \beta\hat{\ }\langle s\rangle$ are eligible to act at the remaining substages of this stage, and these have never acted before and thus will not have a computation $\Phi(x) = 0$. Thus for any $m <$ the current n, at least one of the computations $\Psi(A_i; m)$ will survive from stage s_n to stage s_{n+1}.

3.5. Embedding all countable distributive lattices into the computably enumerable degrees. The minimal pair theorem shows the embeddability of the diamond lattice into the computably enumerable Turing degrees. This result was extended to all finite distributive lattices by Lerman and Thomason (1971), and by Lerman (1971) also to all countable distributive lattices.

Theorem 10. (Lerman (1971)) *Any countable distributive lattice is embeddable into the computably enumerable Turing degrees.*

Proof. By the universality of the countable atomless Boolean algebra $\mathcal{B}$, it suffices to embed $\mathcal{B}$ into $\mathbf{E}$. We represent $\mathcal{B}$ by the collection of finite unions of sets of the form $\{n \cdot 2^m + p \mid n \in \omega\}$ where m ranges over all natural numbers and $p \in [1, 2^m]$. Clearly, $\mathcal{B}$ is a countable atomless Boolean algebra under set-theoretical union and intersection with least element $\emptyset$ and greatest element $\omega - \{0\}$.

For each $i > 0$, we construct a uniformly computably enumerable set A_i; and, for each $S \in \mathcal{B}$, we set $A_S = \{\langle x, i\rangle \mid i \in S \wedge x \in A_i\}$. It is not hard to check (see, e.g., Soare (1987, Section 9.2)) that it suffices to meet, for all Turing functionals Φ and Ψ, all $i \in \omega$, and all $S, T \in \mathcal{B}$ incomparable under set inclusion, the following

Requirements:

$$\mathcal{P}^i_\Phi : A_i \neq \Phi, \text{ and}$$
$$\mathcal{N}^{S,T}_\Psi : \Psi(A_S) = \Psi(A_T) \text{ is total} \to \exists\Delta(\Delta(A_{S\cap T}) = \Psi(A_S)).$$

Here Δ is a partial computable function built by us (i.e., by one of our strategies). (We again use *Posner's Trick*, allowing the same functional for A_S and A_T.)

The *strategy* and *outcomes* for $\mathcal{P}^i_\Phi$ are the same as in the minimal pair theorem.

Strategy for $\mathcal{N}^{S,T}_\Psi$: The strategy builds a "local" partial computable function (i.e., only this strategy may make definitions for Δ whereas other $\mathcal{N}$-strategies define their own versions of Δ).

1. Set $n = 0$.
2. Wait for $\Psi(A_S)\restriction(n+1) = \Psi(A_T)\restriction(n+1)$.
3. Set $\Delta(A_{S\cap T}; m) = \Psi(A_S; m)$ with use $\delta(m) = \max\{\psi(A_S; m), \psi(A_T; m)$ for all $m \leq n$ for which $\Delta(A_{S\cap T}; m)$ is currently undefined. (Note that a previous definition of $\Delta(A_{S\cap T}; m)$ for $m < n$ may have become undefined due to an $A_{S\cap T}$-change since the time it was previously defined.)
4. Preserve either $\Psi(A_S; m)$ or $\Psi(A_T; m)$ (for $m \leq n$) from now on (i.e., when both $\Psi(A_S; m)$ and $\Psi(A_T; m)$ are defined then protect at least one of them, and when only one of them is defined then protect it, by restraining $A_U\restriction(\psi(m)+1)$ for the appropriate $U = S$ or T) until (if ever) $\Delta(A_{S\cap T}; m)$ becomes undefined.
5. Increment n by $+1$ and go back to Step 2.

Outcomes of the $\mathcal{N}$-strategy:

w_n: n is least such that we are eventually waiting at Step 2: Then $\Psi(A_S)\restriction(n+1) \neq \Psi(A_T)\restriction(n+1)$. (These are the finitary outcomes.)

∞: Eventually reach Step 5 for each n: If $\Psi(A_S) = \Psi(A_T)$ is total then clearly so is Δ, and by the restraint, $\Delta(A_{S\cap T})$ must compute their common value correctly. (This is the infinitary outcome.)

We let the current outcome of the $\mathcal{N}$-strategy be ∞ if the strategy has just incremented its parameter n; otherwise the current outcome is w_n for the current value of n. Then a true finitary outcome of the $\mathcal{N}$-strategy is the current outcome at cofinitely many stages while a true infinitary outcome is the current outcome only at infinitely many stages.

We let $\Lambda = \{\infty <_\Lambda \ldots <_\Lambda w_2 <_\Lambda w_1 <_\Lambda w_0 <_\Lambda s <_\Lambda w\}$ be the set of outcomes.

Tree of strategies: Effectively order the requirements (of order type ω). Inductively define a tree $T \subseteq \Lambda^{<\omega}$ by assigning to all strategies α of length e the eth requirement in this list and by letting $\alpha\hat{\ }\langle o\rangle$ be the immediate successors of α where o ranges over all possible outcomes of α.

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$.

Verification: Denote by $f \in [T]$ the true path of the construction, i.e., the liminf of f_s as s tends to infinity. Note that by the above remarks about current outcomes and by the way we ordered the outcomes, f will be an infinite path through T.

Lemma. *Every strategy $\alpha \subset f$ ensures the satisfaction of its requirement.*

Proof. Fix $s_0 \geq |\alpha|$ least such that $\alpha \subseteq f_{s_0}$. Then $\alpha \leq f_s$ for all $s \geq s_0$ by the Π_2-Lemma.

We now distinguish cases for α.

Case 1: α is a $\mathcal{P}^i_\Phi$-strategy: Then α picks a permanent witness x at stage s_0. Now α ensures its requirement as in the basic module above.

Case 2: α is an $\mathcal{N}^{S,T}_\Psi$-strategy: Assume that $\Psi(A_S) = \Psi(A_T)$ is total. Then clearly α has outcome ∞, and $\Delta(A_{S\cap T})$ must be total since each use is bounded by the maximum of the corresponding Ψ-uses, and Δ is redefined whenever needed. It thus remains to establish the correctness of Δ. We will do so by showing that if $\Delta(A_{S\cap T}; m)$ is defined at a stage then it equals at least one of $\Psi(A_S; m)$ and $\Psi(A_T; m)$ at that stage.

Let $s_n \geq s_0$ be the stage at which α first sets its counting parameter to n. We first note (using the Π_2-Lemma) that at any stage $s \geq s_n$, only a $\mathcal{P}$-strategy $\beta \supseteq \alpha\hat{\ }\langle\infty\rangle$ can possibly destroy a computation $\Psi(A_U; m)$ (for $m < n$ and $U = S$ or T) that has existed at stage s_n since all $\beta < \alpha$ no longer enumerate numbers, and all $\beta >_L \alpha\hat{\ }\langle\infty\rangle$ which act at or after stage s_n have witnesses that are too large. But at any stage, at most one number is enumerated into at most one set A_i since once a $\mathcal{P}$-strategy β enumerates a number, only strategies $\beta' \supseteq \beta\hat{\ }\langle s\rangle$ are eligible to act at the remaining substages of this stage, and these have never acted before and thus will not have a computation $\Phi(x) = 0$. Thus for any $m <$ the current n, at least one of the computations $\Psi(A_U; m)$ will survive from stage s_n to stage s_{n+1} unless a $\mathcal{P}$-strategy $\beta \supseteq \alpha\hat{\ }\langle\infty\rangle$ enumerates at a stage s_n a number $x \leq \min\{\psi(A_S; m), \psi(A_T; m)\}$ (for some $m < n$) into the set $A_{S\cap T}$. But then $x \leq \delta(m)$, thus destroying the computation $\Delta(A_{S\cap T}; m)$ as desired, as we will show in the following

Claim.

(i) *If two $\mathcal{P}$-strategies β and β' enumerate witnesses x and x' at stages $s < s'$, respectively, then $x' < x$ or $s < x'$.*

(ii) *For any stage s_n and any $m < n$, there is no $\mathcal{P}$-strategy $\beta \supseteq \alpha\hat{\ }\langle\infty\rangle$ which enumerates a witness x in the interval*

$$(\delta(m)[s_n], \min\{\psi(A_S; m)[s_n], \psi(A_T; m)[s_n]\}]$$

at stage s_n. (Here a use is set to ∞ if the corresponding computation is undefined.)

The interval

$$(\delta(m)[s_n], \min\{\psi(A_S; m)[s_n], \psi(A_T; m)[s_n]\}]$$

in the above claim is called a *dangerous interval* for $\Delta(A_{S\cap T})$ since a number x in this interval entering $A_{S\cap T}$ will destroy both $\Psi(A_S; n)$ and $\Psi(A_T; n)$ while leaving $\Delta(A_{S\cap T}; n)$ intact, allowing the common value of $\Psi(A_S; m)$ and $\Psi(A_T; m)$ to change while $\Delta(A_{S\cap T}; m)$ will still have to be destroyed by a yet smaller number (namely, a number $\leq \delta(m)$) to enter $A_{S\cap T}$. Preventing numbers from entering dangerous intervals is at the heart of current research in lattice embeddings into the computably enumerable degrees.

Proof of claim. (i) For the sake of a contradiction, assume that $x \leq x' \leq s$. Fix the stages t and t' at which x and x' were picked. Then $t \leq t' < s$. Now $\beta \subset \beta'$ is impossible since β permanently changes current outcome at stage s. Also, $\beta' \subset \beta$ is impossible since then β' would choose its witness first and so $x' < x$. Next, $\beta' <_L \beta$ is impossible by the Π_2-Lemma (i) and (ii) since then β' and β have different guesses about the outcome of some strategy $\subset \beta', \beta$, and β guesses a finitary outcome for β. Finally, $\beta <_L \beta'$ is impossible by similar reasoning.

(ii) For the sake of a contradiction, suppose there is such a stage. Let s_l be the least stage at which $x \in (\delta(m)[s_l], \min\{\psi(A_S; m)[s_l], \psi(A_T; m)[s_l]\}]$ for a witness x which is later enumerated. Then at stage s_l, one of $\Psi(A_S; m)$ and $\Psi(A_T; m)$ (say, the latter) is destroyed by a witness x' enumerated by a strategy $\beta' \supseteq \alpha\hat{\ }\langle\infty\rangle$, say. By (i), we conclude that $x < x'$ or $x > s_l$. But the former contradicts $x > \delta(m)[s_n]$ and the minimality of s_l, and the latter implies $x > s_l \geq \psi(A_S; m)[s_l]$, contradicting the choice of s_l. This establishes the claim, the lemma, and the theorem.

3.6. Embedding nondistributive lattices into the computably enumerable degrees. Once the embeddability of all finite distributive lattices was established, attention turned to nondistributive lattices. An easy fact from lattice theory tells us that any nondistributive lattice contains a copy of one of the two nondistributive five-element lattices, N_5 and M_3. (For the latter, we choose to use the notation favored by lattice theorists rather than M_5, the notation frequently used by computability theorists.) Lachlan was able to establish the embeddability of these two lattices. All currently known techniques in lattice embeddings into the computably enumerable degrees are combinatorial variations of the techniques used in the embeddings of these two lattices (as well as the techniques for embedding distributive lattices covered in the previous section).

Theorem 11. (Lachlan (1972)) *The two nondistributive five-element lattices N_5 and M_3 are embeddable into the computably enumerable Turing degrees.*

From the point of view of the construction, the main difference between embeddings of finite distributive and of finite nondistributive lattices is that in the latter, a join-irreducible element a can be below the join of all the elements $\not\leq a$. This causes problems since enumerating a number into the corresponding set A requires the simultaneous enumeration of markers into one of B and C for any corresponding elements a, b, c with $a \leq b \vee c$.

We prove the above two embeddability results separately.

Proof for N_5. We denote the elements of N_5 by 0, a, b, c, and 1, where a and c are the atoms of N_5 and $a < b < a \vee c = 1$. Corresponding to the lattice elements a, b, and c, we construct three computably enumerable sets A, B, and C as well as a Turing functional Γ, meeting, for all Turing functionals Φ and Ψ, the following

Requirements:

$$\begin{aligned} \mathcal{R} &: B = \Gamma(A \oplus C), \\ \mathcal{P}^X_\Phi &: X \neq \Phi \text{ (for } X = A \text{ or } C), \\ \mathcal{P}^B_\Phi &: B \neq \Phi(A), \text{ and} \\ \mathcal{N}_\Psi &: \Psi(A \oplus B) = \Psi(C) \text{ is total} \to \exists \Delta(\Delta = \Psi(A \oplus B)). \end{aligned}$$

Here Δ is a partial computable function built by us (i.e., by one of our strategies). (Here, we set $\mathbf{b} = \deg(A \oplus B)$ and use *Posner's Trick* as usual for these arguments, allowing the same functional Ψ for both $A \oplus B$ and C.)

Strategy for $\mathcal{R}$: At each stage s, we (re)define $\Gamma(A \oplus C; x) = B(x)$ for all $x \leq s$ if needed. Furthermore, we allow the use $\gamma(x)$ to be lifted at most finitely often for each x. Finally, we require that whenever some number x enters B there is a simultaneous $(A \oplus C) \upharpoonright (\gamma(x)+1)$-change. (Recall here that the use $\gamma(x)$ is computed separately, i.e., $\gamma(x)$ is the largest number y such that changing A or C at y will make $\Gamma(A \oplus C; x)$ undefined.)

Strategy for $\mathcal{P}^X_\Phi$ (for $X = A$, B, or C): For $X = A$ or C, we can adopt the same strategy as in the minimal pair argument since there is no need to correct Γ when enumerating into A or C. The situation is different for B. Here, when enumerating a witness x into B, we need to simultaneously enumerate a number $\leq \gamma(x)$ into A or C. Enumerating into C may cause problems for minimal pair strategies whereas enumerating into A might injure the computation $\Phi(A; x)$ we are trying to preserve. We circumvent this problem by first lifting the use $\gamma(x)$ before enumerating the witness x into B as follows:

1. Pick an unused witness x (targeted for B) larger than any number mentioned so far in the construction and keep x out of B.
2. Wait for $\Phi(A; x) = 0$.
3. Restrain $A \upharpoonright (\varphi(x) + 1)$, enumerate $\gamma(x)$ into C, and require that the new use $\gamma(x)$ be greater than $\varphi(x)$.
4. Later (at the next stage at which the strategy is eligible to act), enumerate x into B and the new use $\gamma(x)$ into A and stop.

Outcomes of the $\mathcal{P}^X$-strategy:

- w: Wait at Step 2 forever: Then $X(x) = 0 \neq \Phi(x)$ or $\Phi(A; x)$, respectively.
- s: Stop at the last step: Then $X(x) = 1 \neq 0 = \Phi(x)$ or $\Phi(A; x)$, respectively. (Note that once a $\mathcal{P}^B$-strategy reaches Step 3 it must eventually go on to Step 4 if it is eligible to act again.)

We let the current outcome of a $\mathcal{P}^X$-strategy be w or s depending on whether it is still waiting at Step 2 or has already stopped. A $\mathcal{P}^B$-strategy that has just reached Step 3 has no current outcome until it stops at Step 4. The true outcome of the $\mathcal{P}$-strategy is the current outcome at cofinitely many stages since the delay at Step 3 takes only one stage at which the strategy is eligible to act.

The *strategy* and *outcomes* for $\mathcal{N}_\Psi$ are the same as in the minimal pair argument.

We let $\Lambda = \{\infty <_\Lambda \dots <_\Lambda w_2 <_\Lambda w_1 <_\Lambda w_0 <_\Lambda s <_\Lambda w\}$ be the set of outcomes.

Tree of strategies: Effectively order all $\mathcal{P}$- and $\mathcal{N}$-requirements (of order type ω). Inductively define a tree $T \subseteq \Lambda^{<\omega}$ by assigning to all strategies α of length e the eth requirement in this list and by letting $\alpha\hat{\ }\langle o\rangle$ be the immediate successors of α where o ranges over all possible outcomes of α. (Requirement $\mathcal{R}$ is a global requirement which will not be assigned to any node on the tree but will be handled separately at the end of each stage.)

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$. If a $\mathcal{P}^B$-strategy α is eligible to act at stage s and reaches Step 3 at that stage then we end the stage by not allowing substages $> |\alpha|$ at stage s and setting the current true path $f_s = \alpha$.

At the end of each stage s, (re)define $\Gamma(A \oplus C; x)$ for all $x \leq s$ (for which $\Gamma(A \oplus C; x)$ is currently undefined) with the old use $\gamma(x)$ (if previously defined and not required to be lifted at stage s by any $\mathcal{P}^B$-strategy) or with use $\gamma(x)$ larger than any number mentioned so far in the construction (otherwise).

Verification: Denote by $f \in [T]$ the true path of the construction, i.e., the lim inf of f_s as s tends to infinity. Note that by the above remarks about current outcomes, f will be an infinite path through T, i.e., f_s cannot infinitely often be of fixed finite length.

Lemma 1. *Requirement $\mathcal{R}$ is met.*

Proof. The construction never allows a number x to enter B without a simultaneous $(A \oplus C)\restriction(\gamma(x)+1)$-change. Furthermore, the use $\gamma(x)$ is lifted at most once for each x since once it is lifted, x remains the witness of its strategy (and thus cannot be lifted again).

Lemma 2. *Every strategy $\alpha \subset f$ ensures the satisfaction of its requirement.*

Proof. Fix s_0 least such that $\alpha \subseteq f_{s_0}$. Then $\alpha \leq f_s$ for all $s \geq s_0$ by the Π_2-Lemma.

We now distinguish cases for α.

Case 1: α is a $\mathcal{P}^X_\Phi$-strategy (for $X = A$, B, or C): Then α picks a permanent witness x at stage s_0. Now α enumerates x into X iff it ever finds a computation $\Phi(x) = 0$ or $\Phi(A; x) = 0$. If $X = A$ or C, or if $x \notin X$, then the lemma is clear. So suppose $X = B$ and α finds a computation $\Phi(A; x) = 0$ at a stage $s_1 \geq s_0$. Then by hypothesis on s_0, no $\beta < \alpha$ will enumerate any number after stage s_1; no $\beta >_L \alpha\hat{\ }\langle s\rangle$ which acts at or after stage s_1 can enumerate any number $\leq \varphi(x)$ into any set; and any $\beta \supseteq \alpha\hat{\ }\langle s\rangle$ cannot act before stage s_1 and can thus not even pick a number $\leq \varphi(x)$.

Case 2: α is an $\mathcal{N}_\Psi$-strategy: Assume that $\Psi(A \oplus B) = \Psi(C)$ is total. Then clearly α has outcome ∞, and so Δ is total. It thus remains to verify the correctness of Δ. We will show this by establishing that once $\Delta(n)$ is defined, it equals at least one of $\Psi(A \oplus B; n)$ and $\Psi(C; n)$ at any stage.

Let $s_n \geq s_0$ be the stage at which α first sets its counting parameter to n. We first note that at any stage $s \geq s_n$, only a $\mathcal{P}$-strategy $\beta \supseteq \alpha\hat{\ }\langle\infty\rangle$ can possibly destroy a computation $\Psi(A \oplus B; m)[s_n]$ or $\Psi(C; m)[s_n]$ (for $m < n$) that has existed at stage s_n since all $\beta < \alpha$ no longer enumerate numbers, and, all $\beta >_L \alpha\hat{\ }\langle\infty\rangle$ which act at or after stage s_n have witnesses that are too large. But at any stage, numbers are enumerated only into A and B, or only into C, since once a $\mathcal{P}$-strategy β enumerates a number, the stage is either ended, or only strategies $\beta' \supseteq \beta\hat{\ }\langle s\rangle$ are eligible to act at the remaining substages of this stage and these have never acted before and thus will not have a computation $\Phi(x) = 0$ or $\Phi(A; x) = 0$. Thus for any $m <$ the current n, at least one of the computations $\Psi(A \oplus B; m)$ or $\Psi(C; m)$ will survive from stage s_n to stage s_{n+1}.

This establishes the embeddability of N_5.

The proof for M_3 is quite a bit harder. For N_5, we were able to show that at worst a two-step process of enumerating numbers into sets can ensure the satisfaction of a diagonalization requirement. For M_3, there is no such fixed bound on the number of steps; rather, the number of steps required to eventually enumerate a fixed witness x into a set depends on when the corresponding Φ-computation against we wish to diagonalize is found, and on the subsequent behavior of higher-priority minimal pair strategies.

Proof for M_3. We denote the atoms of the lattice M_3 by a_0, a_1, and a_2, and assume that for any two distinct such atoms, the meet is 0 and the join is 1. We then construct three computably enumerable sets A_0, A_1, and A_2 and three Turing functionals Γ_0, Γ_1, and Γ_2, meeting, for all pairwise distinct $i, j, k \leq 2$ and all Turing functionals Φ and Ψ, the following

Requirements:

$$\begin{aligned}
\mathcal{R}^i &: A_i = \Gamma_i(A_j \oplus A_k) \text{ (for } j < k),\\
\mathcal{P}^{i,j}_\Phi &: A_i \neq \Phi(A_j), \text{ and}\\
\mathcal{N}^{i,j}_\Psi &: \Psi(A_i) = \Psi(A_j) \text{ is total} \to \exists\Delta(\Delta = \Psi(A_i)) \text{ (for } i < j).
\end{aligned}$$

Here Δ is a partial computable function built by us (i.e., by one of our strategies). (We use *Posner's Trick* as usual for these arguments, allowing the same functional Ψ for both A_i and A_j.)

Strategy for $\mathcal{R}^i$: At each stage s, we (re)define $\Gamma_i(A_j \oplus A_k; x) = A_i(x)$ for all $x \leq s$ if needed. Furthermore, we allow the use $\gamma_i(x)$ to be lifted at most finitely often for each x, namely, whenever $\gamma_i(x)$ is enumerated into A_j or A_k. This will make the Γ's total.

The minimal condition for ensuring the correctness of all Γ's is that whenever some number x enters A_i then there is a simultaneous $(A_j \oplus A_k)\restriction(\gamma_i(x) + 1)$-change. (Recall here that the use $\gamma_i(x) > x$ is computed separately, i.e., $\gamma_i(x)$ is the largest number y such that changing A_j or A_k at y will make $\Gamma_i(A_j \oplus A_k; x)$ undefined.) Note that this will also allow the correction of $\Gamma_j(A_i \oplus A_k; \gamma_i(x))$ or $\Gamma_k(A_i \oplus A_j; \gamma_i(x))$, respectively, due to the $A_i(x)$-change.

However, we choose to use a slightly redundant strategy for $\mathcal{R}^i$ since a number and its use frequently become separated in the construction below. Given distinct $i, j \leq 2$, a number x targeted for A_i, and a stage s, we call the (i, j, s)-*sequence of*

x the finite sequence

$$x,\ \gamma_i(x)[s],\ \gamma_j\gamma_i(x)[s],\ \gamma_i\gamma_j\gamma_i(x)[s], \dots,\ \gamma_l\cdots\gamma_j\gamma_i(x)[s]$$

(for $l = i$ or j such that $\Gamma_{i+j-l}(A_l \oplus A_k; \gamma_l \cdots \gamma_j\gamma_i(x))$ is undefined at stage s). We impose on the construction that whenever we *consider*, at some stage s, enumerating into a set A_i a number x (as a witness or the use of another number to be enumerated) and this number x is enumerated at a stage $s' \geq s$, then between stages s and s' (for some $j \neq i$) we will have enumerated its entire (i,j,s)-sequence into the sets A_i and A_j, respectively, either simultaneously or in reverse order. Since this meets the minimal condition mentioned in the last paragraph, it should now be clear that this restriction on enumeration ensures that all Γ's are correct.

Strategy for $\mathcal{P}^{i,j}_{\Phi}$: We adopt a modified Friedberg-Muchnik strategy similar to the one used in the proof for N_5, observing the above restriction on enumerating numbers into A_i. Fix $k \leq 2$ such that $k \neq i,j$. When we are ready to enumerate a witness x into A_i at a stage s then we also enumerate its (i,k,s)-sequence into A_i and A_k, respectively. (We cannot use A_j-enumeration due to the strategy's A_j-restraint.) We thus proceed as follows:

1. Pick an unused witness x (targeted for A_i) larger than any number mentioned so far in the construction and keep x out of A_i.
2. Wait for $\Phi(A_j; x) = 0$ (at a stage s, say).
3. Restrain $A_j \restriction (\varphi(x) + 1)$.
4. Ensure the enumeration of x's (i,k,s)-sequence into A_i and A_k, respectively. (In the full construction later on, let β be the longest $\mathcal{N}$-strategy of which the $\mathcal{P}$-strategy guesses the infinite outcome. If β does not exist then the $\mathcal{P}$-strategy can enumerate the sequence immediately; otherwise β will *process* the sequence next as described below. It may take a finite number of stages until the entire sequence has been enumerated.)
5. When x has been enumerated into A_i then stop.

Outcomes of the $\mathcal{P}$-strategy:

w: Wait at Step 2 forever: Then $A_i(x) = 0 \neq \Phi(A_j; x)$.
s: Stop at Step 5: Then $A_i(x) = 1 \neq 0 = \Phi(A_j; x)$.

(We will show later that once a $\mathcal{P}$-strategy reaches Step 3, x will have been enumerated by the time the $\mathcal{P}$-strategy is eligible to act again.)

We let the current outcome of a $\mathcal{P}$-strategy be w or s depending on whether it is still waiting at Step 2 or has already stopped at Step 5. A $\mathcal{P}$-strategy that has reached Step 3 but not yet stopped at Step 5 has no current outcome until it stops at Step 5. The true outcome of the $\mathcal{P}$-strategy is the current outcome at cofinitely many stages since the delay in Step 4 only lasts until the $\mathcal{P}$-strategy is eligible to act next.

Strategy for $\mathcal{N}^{i,j}_{\Psi}$: The strategy builds a "local" partial computable function (i.e., only this strategy may make definitions for Δ whereas other $\mathcal{N}$-strategies define their own versions of Δ). Fix $k \leq 2$ such that $k \neq i,j$. Then the $\mathcal{N}$-strategy can allow (l,m,s)-sequences (where one of l or m equals k) to be enumerated all at once whereas (i,j,s)- or (j,i,s)-sequences must be separated so that only one element at a time (in reverse order) is enumerated so as to allow the injured computation $\Psi(A_i)\restriction(n+1)$ or $\Psi(A_j)\restriction(n+1)$ to recover before the other is destroyed.

1. Set $n = 0$.

2. Wait for $\Psi(A_i)\restriction(n+1) = \Psi(A_j)\restriction(n+1)$.
3. Set $\Delta(n) = \Psi(A_i; n)$, and preserve either $\Psi(A_i; n)$ or $\Psi(A_j; n)$ from now on, i.e., when both $\Psi(A_i; n)$ and $\Psi(A_j; n)$ are defined then protect at least one of them, and when only one of them is defined then protect it, by restraining $A_l\restriction(\psi(n) + 1)$ for the appropriate $l = i$ or j.
4. If, at the current stage s, some other strategy (necessarily one single strategy below the infinite outcome of the $\mathcal{N}$-strategy) sends to the $\mathcal{N}$-strategy the (l, m, s)-sequence of some number x to be processed then the $\mathcal{N}$-strategy ensures the enumeration of this sequence into A_l and A_m before proceeding to Step 5. (In the full construction later on, let β be the longest $\mathcal{N}$-strategy of which our $\mathcal{N}$-strategy guesses the infinite outcome.) We now distinguish four cases:
   4.1. One of l or m equals k, and β does not exist: Then the $\mathcal{N}$-strategy enumerates the entire sequence at once and proceeds to Step 5.
   4.2. One of l or m equals k, and β exists: Then the $\mathcal{N}$-strategy lets β process the entire sequence and proceeds to Step 5.
   4.3. Neither of l and m equals k (i.e., $\{i, j\} = \{l, m\}$, and β does not exist): Then the $\mathcal{N}$-strategy enumerates the elements of the (l, m, s)-sequence of x, in reverse order, one at a time, always waiting for $\Psi(A_i)\restriction(n+1) = \Psi(A_j)\restriction(n+1)$ before proceeding to the next number, and always accompanied by its current (l, k, s')-sequence or (m, k, s')-sequence (at stage s', say). When x has finally been enumerated then the $\mathcal{N}$-strategy proceeds to Step 5.
   4.4. Neither of l and m equals k (i.e., $\{i, j\} = \{l, m\}$, and β exists): Then the $\mathcal{N}$-strategy lets β *process*, one at time, in reverse order, the elements y of the (l, m, s)-sequence of x, always requesting that the γ_l- or γ_m-use enumerated now be lifted large, and always waiting for $\Psi(A_i)\restriction(n+1) = \Psi(A_j)\restriction(n+1)$ before proceeding to the next number, and always accompanied by y's current (l, k, s')-sequence or (m, k, s')-sequence (at stage s', say). Once x is finally processed by β then the $\mathcal{N}$-strategy proceeds to Step 5.

   We note that in Cases 4.1 and 4.2, x's sequence will destroy only one side for the $\mathcal{N}$-strategy whereas in Cases 4.3 and 4.4, the $\mathcal{N}$-strategy must separate the sequence so as to always protect one of $\Psi(A_i)\restriction(n+1)$ and $\Psi(A_j)\restriction(n+1)$. The action by β and the even higher-priority strategies cannot injure the computations protected by the $\mathcal{N}$-strategy since this action only involves large numbers.
5. Increment n by $+1$ and go back to Step 2.

Outcomes of the $\mathcal{N}$-strategy:

w_n: Wait at Step 2 forever for some n: Then $\Psi(A_i)\restriction(n+1) \neq \Psi(A_j)\restriction(n+1)$. (These are finitary outcomes.)

$w_{n,q}$: Wait at Step 4.3 or 4.4 forever for some n and having already enumerated or processed q many numbers at this step: Then $\Psi(A_i)\restriction(n+1) \neq \Psi(A_j)\restriction(n+1)$. (These are again finitary outcomes.)

∞: Eventually reach Step 5 for each n: Then clearly Δ is total. If furthermore $\Psi(A_i) = \Psi(A_j)$ is total, then by the restraint, Δ must compute their common value correctly. (This is the infinitary outcome.)

We let the *current outcome* of the $\mathcal{N}$-strategy be ∞ if the strategy has just incremented its parameter n; n if the strategy is still waiting at Step 2 with parameter n; and (n, q) if the strategy is still waiting at Step 4 with parameter n after having processed or enumerated q many numbers. (∞ is the infinitary outcome, the others are the finitary outcomes.) Then a true finitary outcome of the $\mathcal{N}$-strategy is the current outcome at cofinitely many stages whereas a true infinitary outcome is the true outcome only at infinitely many stages. (Note that the delay in Steps 4.3 and 4.4 may be indefinite since we may wait for $\Psi(A_i)\restriction(n+1) = \Psi(A_j)\restriction(n+1)$ forever.)

We let $\Lambda = \{\infty <_\Lambda \ldots <_\Lambda w_{2,2} <_\Lambda w_{2,1} <_\Lambda w2,0 <_\Lambda w_2 <_\Lambda \ldots <_\Lambda w_{1,2} <_\Lambda w_{1,1} <_\Lambda w1,0 <_\Lambda w_1 <_\Lambda \ldots <_\Lambda w_{0,2} <_\Lambda w0,1 <_\Lambda w_{0,0} <_\Lambda w_0 <_\Lambda s <_\Lambda w\}$ be the set of outcomes.

Tree of strategies: Effectively order all $\mathcal{P}$- and $\mathcal{N}$-requirements (of order type ω). Inductively define a tree $T \subseteq \Lambda^{<\omega}$ by assigning to all strategies α of length e the eth requirement in this list and by letting $\alpha\hat{\ }\langle o\rangle$ be the immediate successors of α where o ranges over all possible outcomes of α. (Requirements $\mathcal{R}^i$ are global requirements which will not be assigned to any node on the tree but will be handled separately at the end of each stage.)

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$. When a $\mathcal{P}$- or $\mathcal{N}$-strategy α wants an $\mathcal{N}$-strategy $\beta \subset \alpha$ to newly process numbers at stage s then we end the stage by not allowing substages $> |\alpha|$ at stage s and setting the current true path $f_s = \alpha$. (β will then start processing these numbers at the next stage $s' > s$ at which $\beta \subseteq f_{s'}$.)

At the end of each stage s, (re)define $\Gamma_i(A_j \oplus A_k; x) = A_i(x)$ for all $x \leq s$ (for which $\Gamma_i(A_j \oplus A_k; x)$ is currently undefined) with the old use $\gamma_i(x)$ (if previously defined and not yet enumerated into A_j or A_k) or with use $\gamma_i(x)$ larger than any number mentioned in the construction so far (otherwise).

Verification: Denote by $f \in [T]$ the true path of the construction, i.e., the lim inf of f_s as s tends to infinity. Note that by the above remarks about current outcomes, f will be an infinite path through T.

Lemma 1. *Each requirement $\mathcal{R}^i$ is met.*

Proof. The construction never allows a number x to enter A_i without its (i, j)- or (i, k)-sequence entering at the same stage, thus causing a simultaneous $(A_j \oplus A_k)\restriction(\gamma_i(x) + 1)$-change if $\Gamma_i(A_j \oplus A_k)$ is defined. Furthermore, the use $\gamma_i(x)$ is lifted at most finitely for each x since whenever it is lifted, x is either enumerated or processed by a strategy of higher priority than at the last time $\gamma_i(x)$ was lifted. Finally note that x can be chosen as a witness or the use of another number at most once.

Lemma 2. *Every strategy $\alpha \subset f$ ensures the satisfaction of its requirement.*

Proof. Fix s_0 least such that $\alpha \subseteq f_{s_0}$. Then $\alpha \leq f_s$ for all $s \geq s_0$ by the Π_2-Lemma.

We now distinguish cases for α.

Case 1: α is a $\mathcal{P}^{i,j}_\Phi$-strategy: Then α picks a permanent witness x at stage s_0. Now α proceeds to Step 4 iff it ever finds a computation $\Phi(A_j; x) = 0$. If the latter fails we are done; else suppose α finds $\Phi(A_j; x) = 0$ at a stage $s_1 \geq s_0$. Then by the way sequences are processed by higher-priority $\mathcal{N}$-strategies, x will be enumerated before α is eligible to act again; since $\alpha \subset f$, this will occur. Furthermore, by hypothesis on s_0, no $\mathcal{N}$-strategy β with $\beta\hat{\ }\langle\infty\rangle <_L \alpha$ and no $\mathcal{P}$-strategy $\beta < \alpha$ will enumerate any number after stage s_1; no $\beta >_L \alpha\hat{\ }\langle s\rangle$ can enumerate any number

$\leq \varphi(x)$ into any set; and any $\beta \supseteq \alpha\hat{\ }\langle s\rangle$ is not eligible to act before stage s_1 and can thus not enumerate a number $\leq \varphi(x)$. So suppose β with $\beta\hat{\ }\langle\infty\rangle \subseteq \alpha$ is an $\mathcal{N}$-strategy. Then β processes numbers arising from $\mathcal{P}$-strategies $\alpha' \supseteq \beta\hat{\ }\langle\infty\rangle$; by hypothesis on s_0 and s_1, only numbers arising from α itself may be small enough to destroy $\Phi(A_j; x)$. But if such a number y enters A_j after stage s_1 then it does so because $y = \gamma_l(y')$ for $l = i$ or k, and the previous use $\gamma_l(y')$ must have entered A_i or A_k, causing $\gamma_l(y')$ to be lifted above $\varphi(x)$. Thus $\Phi(A_j; x)$ cannot be destroyed after stage s_1.

Case 2: α is an $\mathcal{N}^{i,j}_\Psi$-strategy: Assume that $\Psi(A_i) = \Psi(A_j)$ is total. Then clearly α has outcome ∞, and so Δ is total. It thus remains to verify the correctness of Δ. We will show this by establishing that once $\Delta(n)$ is defined, it equals at least one of $\Psi(A_i; n)$ and $\Psi(A_j; n)$ at any stage.

Let $s_n \geq s_0$ be the stage at which α first sets its counting parameter to n. We first note that no $\mathcal{N}$-strategy β with $\beta\hat{\ }\langle\infty\rangle <_L \alpha$ and no $\mathcal{P}$-strategy $\beta < \alpha$ will enumerate any number after stage s_0; and no $\beta >_L \alpha\hat{\ }\langle\infty\rangle$ which acts at or after stage s_n can enumerate any number $\leq \psi(A_i; m)[s_n]$ or $\psi(A_j; m)[s_n]$ (for $m < n$) into any set. Furthermore, any number that a strategy $\supseteq \alpha\hat{\ }\langle\infty\rangle$ wants to enumerate is first processed by α and thus not directly enumerated.

So suppose β is an $\mathcal{N}$-strategy with $\beta\hat{\ }\langle\infty\rangle \subseteq \alpha\hat{\ }\langle\infty\rangle$. Then β processes numbers arising from $\mathcal{P}$-strategies $\alpha' \supseteq \beta\hat{\ }\langle\infty\rangle$; by hypothesis on s_0 and s_n, only numbers arising via $\mathcal{P}$-strategies $\alpha' \supseteq \alpha\hat{\ }\langle\infty\rangle$ can be small enough to destroy $\Psi(A_i; m)[s_n]$ or $\Psi(A_j; m)[s_n]$ (for $m <$ the current n) at or after stage s_n. But if such a number y enters A_i or A_j at or after stage s_n but before stage s_{n+1} then it was either processed by α at stage s_n or part of a new sequence formed by an $\mathcal{N}$-strategy $\subset \alpha$. Any numbers processed by α, by the way α processes numbers, leave one of $\Psi(A_i; m)$ and $\Psi(A_j; m)$ intact until stage s_{n+1}; and any number y created by an $\mathcal{N}$-strategy $\subset \alpha$ must be of the form $y = \gamma_l(y')$ for some l, and the previous use $\gamma_l(y')$ must have entered at or after stage s_n, causing $\gamma_l(y')$ to be lifted above both $\psi(A_i; m)[s_n]$ and $\psi(A_j; m)[s_n]$.

Thus for any $m <$ the current n, at least one of the computations $\Psi(A_i; m)$ or $\Psi(A_j; m)$ will survive from stage s_n to stage s_{n+1}.

This establishes the embeddability of M_3.

3.7. A nonbranching degree. The following construction uses, strictly speaking, Π_3-strategies; however, since every strategy affects the others only in a finitary way, the construction resembles a finite injury priority argument. We compromise by presenting it here.

While the Sacks Splitting Theorem shows that every nonzero computably enumerable degree is join-reducible, the following theorem shows that not every incomplete computably enumerable degree is meet-reducible.

Theorem 12. (Lachlan (1966)) *There is a* nonbranching *degree, i.e., an incomplete computably enumerable Turing degree which is not the infimum of two greater computably enumerable degrees.*

Proof. We construct computably enumerable sets A and C, meeting, for all computably enumerable sets W_0 and W_1 and all Turing functionals Φ and Ψ, the following

Requirements:

$$\begin{aligned}\mathcal{N}_\Phi &: C \neq \Phi(A),\\ \mathcal{R}_{W_0,W_1} &: \exists B \exists \Gamma_0 \exists \Gamma_1 (B = \Gamma_0(A \oplus W_0) = \Gamma_1(A \oplus W_1)), \text{ and}\\ \mathcal{R}_{W_0,W_1,\Psi} &: B = \Psi(A) \to \exists \Lambda_0 (W_0 = \Lambda_0(A)) \vee \exists \Lambda_1 (W_1 = \Lambda_1(A)).\end{aligned}$$

Here the Γ's and Λ's are partial computable functionals built by us (i.e., by one of our strategies).

Strategy for $\mathcal{N}_\Phi$: This is the Π_1-strategy we have already encountered in the Friedberg-Muchnik Theorem. (The set C is merely "our version" of the complete set K.) The outcomes are w and s.

Strategy for $\mathcal{R}_{W_0,W_1}$: The strategy builds a "local" computably enumerable set B and "local" partial computable functionals Γ_0 and Γ_1 (i.e., only this strategy may make definitions for Γ_0 and Γ_1, and only this strategy and its $\mathcal{R}_{W_0,W_1,\Psi}$-substrategies will use this set and these functionals).

At stage s, the strategy first defines $\Gamma_0(A \oplus W_0; s)$ and $\Gamma_1(A \oplus W_1; s)$ with a use larger than any number mentioned so far in the construction. At subsequent stages $s' > s$, the strategy redefines these (if they have become undefined through $A \oplus W_i$-changes), increasing the use (to a number larger than any number mentioned so far in the construction) at most finitely often when requested by one of its $\mathcal{R}_{W_0,W_1,\Psi}$-substrategies. The strategy also allows its $\mathcal{R}_{W_0,W_1,\Psi}$-substrategies to enumerate a number x into B only when both $\Gamma_0(A \oplus W_0; x)$ and $\Gamma_1(A \oplus W_1; x)$ are undefined.

Since the requirement has no hypothesis, we do not distinguish outcomes for this strategy.

Strategy for $\mathcal{R}_{W_0,W_1,\Psi}$: The strategy builds "local" partial computable functionals Λ_0 and $\Lambda_{1,n}$ for $n \in \omega$ (i.e., only this strategy may make definitions for Λ_0 and $\Lambda_{1,n}$).

The strategy consists of "cycles" (n_0, n_1) acting in a coordinated fashion, with cycle $(0,0)$ starting first. (Cycles (n_0, n_1) (for all $n_1 \in \omega$) will jointly define functional Λ_{1,n_0}, whereas all cycles will jointly define Λ_0.) Cycle (n_0, n_1) proceeds as follows:

1. Pick an unused witness $x = x_{n_0,n_1}$ (targeted for B) larger than any number mentioned thus far in the construction and keep x out of B.
2. Wait for $\Psi(A; x) = 0$.
3. Set $\Lambda_{1,n_0}(A; n_1) = W_1(n_1)$ with use $\lambda_{1,n_0}(n_1)$ equal to the greatest of $\psi(x)$, $\lambda_{1,n_0}(n_1 - 1)$, and $\lambda_0(n_0 - 1)$. (The latter two keep the uses nondecreasing in n_0 and n_1; for convenience, we set $\lambda_{1,n_0}(-1) = \lambda_0(-1) = 0$.)
4. Start cycle $(n_0, n_1 + 1)$. From now on, if $A \restriction (\lambda_{1,n_0}(n_1) + 1)$ changes then cancel all cycles $> (n_0, n_1)$ (under the lexicographical ordering) and return to Step 2. (Note that we do *not* impose A-restraint. If the cycles $> (n_0, n_1)$ are canceled then the corresponding A-change will undo their definitions for the Λ's.)
5. Wait for n_1 to enter W_1.
6. Stop all cycles $> (n_0, n_1)$ and request that $\gamma_1(x)$ be increased above $\psi(x)$ when $\Gamma_1(A \oplus W_1; x)$ is redefined.
7. Set $\Lambda_0(A; n_0) = W_0(n_0)$ with use $\lambda_0(n_0) = \lambda_{1,n_0}(n_1)$.
8. Start cycle $(n_0 + 1, 0)$.
9. Wait for n_0 to enter W_0.

10. Stop all cycles $> (n_0, n_1)$ and request that $\gamma_0(x)$ be increased above $\psi(x)$ when $\Gamma_0(A \oplus W_0; x)$ is redefined.
11. Enumerate x into B and $\gamma_0(x)$ and $\gamma_1(x)$ into A, restrain $A \upharpoonright (\psi(x)+1)$, and stop all cycles.

Outcomes of the $\mathcal{N}$-strategy: There are a variety of finitary and infinitary outcomes:

(n_0, n_1, w): Wait at Step 2 forever for some fixed (n_0, n_1): Then $\Psi(A; x) \neq B(x)$, a finitary outcome.

(n_0, n_1, ∞): Return to Step 2 via Step 4 infinitely often for some fixed (n_0, n_1): Then $\Psi(A; x)$ is undefined, an infinitary outcome.

(n_0, ∞): Wait at Step 5 forever for some fixed n_0 and all n_1: Then $\Lambda_{1,n_0}(A) = W_1$, an infinitary outcome.

∞: Wait at Step 9 forever for all n_0 (each for some fixed n_1): Then $\Lambda_0(A) = W_0$, an infinitary outcome.

s: Reach Step 11 for some (n_0, n_1): Then $\Psi(A; x) = 0 \neq 1 = B(x)$.

These outcomes are thus quite complicated and cannot be incorporated into a Π_2-priority argument tree. But note that the only action affecting other strategies is the enumeration into A and B at Step 11 when the strategy stops. So for the purposes of the tree, we will collapse all of the non-s-outcomes into one "wait" outcome w. The *current outcome* of the strategy is w if the strategy has not yet reached Step 11, and s otherwise. In terms of outcomes, this will make the strategy behave like a Π_1-strategy.

We fix $\Lambda = \{s, w\}$ as the set of outcomes.

Tree of strategies: The $\mathcal{R}_{W_0,W_1}$-strategies are global and will thus not be put on the tree. We fix an effective listing (of order type ω) of these global requirements. We also effectively order the $\mathcal{N}_\Phi$- and $\mathcal{R}_{W_0,W_1,\Psi}$-requirements (of order type ω). On the tree $T = \Lambda^{<\omega}$, we assign to all strategies α of length e the eth requirement in the latter list.

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$. At the end of each stage s, the first s many $\mathcal{R}_{W_0,W_1}$-strategies act.

Verification:

Lemma 1. *Every $\mathcal{R}_{W_0,W_1}$-requirement is satisfied.*

Proof. For each x, the $\mathcal{R}_{W_0,W_1}$-strategy will redefine $\Gamma_i(A \oplus W_i; x)$ at all but finitely many stages at which it is undefined. The use can be increased only if x is some witness x_{n_0,n_1} of an $\mathcal{R}_{W_0,W_1,\Psi}$-strategy, and then only once, namely just after n_i enters W_i. Thus the Γ's must be total. By the A-enumeration in Step 11, the Γ's must also be correct.

Lemma 2. *Every strategy $\alpha \subset f$ ensures the satisfaction of its requirement.*

Proof. Fix s_0 least such that $\alpha \subseteq f_{s_0}$ (and so $\alpha \subseteq f_s$ for all $s \geq s_0$ by the Π_1-Lemma).

We distinguish cases by the type of requirement α works on:

Case 1: α is an $\mathcal{N}_\Phi$-strategy: Then α picks a witness x at stage s_0. Then α either never enumerates x (in which case $C(x) = 0 \neq \Phi(A; x)$); or α enumerates x at a stage $s_1 \geq s_0$, and so $C(x) = 1 \neq 0 = \Phi(A; x)[s_1]$. As in the case of the Friedberg-Muchnik Theorem, we can now argue that no strategy can enumerate a number $\leq \varphi(x)$ into A after stage s_1.

Case 2: α is an $\mathcal{R}_{W_0,W_1,\Psi}$-strategy: Then α works just as in the description above. Note that the only potential injury to α's action is due to A-injury after α has reached Step 11. But then α has changed outcome, so we can argue as in Case 1.

3.8. The Lachlan Splitting Theorem. The Lachlan Splitting Theorem refines the Sacks Splitting Theorem in the sense that it ensures in addition that the Turing degrees of the two splitting halves have an infimum.

The construction for this theorem, like that of the Sacks Splitting Theorem, uses a priority ordering of the requirements that is not arbitrary and must differ for different paths through the priority tree. In addition, in this argument, we have action "off the true path", i.e., by nodes which are not currently on the true path. This is necessary since we need to make use of W-permissions immediately.

Theorem 13. (Lachlan (1980)) *For any computably enumerable set $W >_T \emptyset$ there are computably enumerable sets A_0 and A_1 of incomparable Turing degree such that W is the disjoint union of A_0 and A_1 and such that the infimum of the degrees of A_0 and A_1 exists.*

Remark. Indeed the sets A_0, A_1, and B (of Turing degree $\deg A_0 \cap \deg A_1$) are all wtt-reducible to W.

Proof. We need to construct computably enumerable set A_0, A_1, and B as well as a functional Γ, meeting, for all $x \in \omega$, all $i \leq 1$, and all Turing functionals Φ and Ψ, the following

Requirements:

$$\begin{aligned} \mathcal{R} &: A_0, A_1 \subseteq W \wedge A_0 \cap A_1 = \emptyset, \\ \mathcal{G} &: B = \Gamma(W), \\ \mathcal{P}_x &: x \in W \to x \in A_0 \cup A_1, \\ \mathcal{N}^i_\Phi &: W = \Phi(A_i \oplus B) \to \exists \Delta (W = \Delta), \text{ and} \\ \mathcal{M}_\Psi &: \Psi(A_0) = \Psi(A_1) \text{ is total} \to \exists \Theta(\Theta(B) = \Psi(A_0 \oplus B)). \end{aligned}$$

Here Δ and Θ are partial computable functionals built by us (i.e., by one of our strategies).

The *strategies* for $\mathcal{R}$, $\mathcal{P}_x$, and $\mathcal{N}^i_\Phi$ are the same as in the Sacks Splitting Theorem, with the following important exception: In the present argument, the $\mathcal{N}^i$-strategies will impose an even higher A_i-restraint (namely, equal to the stage s at which the restraint is imposed) in order to protect against injury by the $\mathcal{M}$-strategies. This will ensure that if the $\mathcal{N}^i$-strategy is injured by a higher-priority $\mathcal{M}$-strategy's enumerating $\vartheta(x) \leq s$ into B to correct $\Theta(B; x)$ then this happens only after $(A_i \oplus B) \restriction (s+1)$ has already changed before, thus injuring the $\mathcal{N}^i$-strategy's restraint before. This will allow us to conclude later that an $\mathcal{N}$-strategy cannot be injured unless it is also initialized.

Strategy for $\mathcal{G}$: The strategy builds a "global" functional Γ by defining, at the end of each stage s, $\Gamma(W; x) = B(x)$ (for all $x \leq s$ for which $\Gamma(W; x)$ is currently undefined) with use $\gamma(x) = x$. We will show below (in Lemma 4) that the $\mathcal{M}$-strategies do not enumerate any number x into B unless $\Gamma(B; x)$ is currently undefined.

Strategy for $\mathcal{M}_\Psi$: The strategy builds a "local" functional (i.e., only this strategy may make definitions for Θ whereas other $\mathcal{M}$-strategies define their own versions of Θ).

1. Set $n = 0$.
2. Wait for $\Psi(A_0 \oplus B)\restriction(n+1) = \Psi(A_1 \oplus B)\restriction(n+1)$.
3. Set $\Theta(B; y) = \Psi(A_0 \oplus B; y)$ (for all $y \leq n$ for which $\Theta(B; y)$ is currently undefined) with use equal to the greater of $\psi(A_0 \oplus B; y)$ and $\psi(A_1 \oplus B; y)$; and try to preserve either $\Psi(A_0 \oplus B; n)$ or $\Psi(A_1 \oplus B; n)$ from now on, i.e., when both $\Psi(A_0 \oplus B; n)$ and $\Psi(A_1 \oplus B; n)$ are defined then try to protect at least one of them, and when only one of them is defined then try to protect it, by restraining $(A_i \oplus B)\restriction(\psi(n)+1)$ for the appropriate i.
4. Restart at Step 2 with $n+1$ in place of n. From now on, whenever $\Theta(B; n)$ is defined but equals neither $\Psi(A_0 \oplus B; n)$ nor $\Psi(A_1 \oplus B; n)$, then enumerate $\vartheta(n)$ into B immediately in order to destroy $\Theta(B; n)$. Whenever $\Theta(B; n)$ is undefined, return to Step 2.

Outcomes of the $\mathcal{M}$-strategy:

- s: Wait at Step 2 forever for some n, starting at a stage s: Then $\Psi(A_0 \oplus B)\restriction(n+1) \neq \Psi(A_1 \oplus B)\restriction(n+1)$. (These are the finitary outcomes.)
- ∞: Eventually reach Step 4 for each n: Then if $\Psi(A_0 \oplus B) = \Psi(A_1 \oplus B)$ is total, so is $\Theta(B)$, since its uses are bounded by the greater of the two Ψ-uses, and by Step 4, $\Theta(B)$ correctly computes $\Psi(A_0 \oplus B)$. (This is the infinitary outcome.)

We let the *current outcome* of the $\mathcal{M}$-strategy be ∞ if the strategy has just incremented its parameter n; otherwise the current outcome is the stage s at which n was last incremented. (I.e., we modify the outcome as for the $\mathcal{M}$-strategies and for the same reasons.) We also define the *current finitary outcome* of the $\mathcal{M}$-strategy to be the greatest stage $s \leq$ the current stage at which n has been incremented.

We let $\Lambda = \{\infty <_\Lambda \ldots <_\Lambda 2 <_\Lambda 1 <_\Lambda 0 <_\Lambda s <_\Lambda w\}$ be the set of outcomes.

Tree of strategies: As for the Sacks Splitting Theorem, we need to use a dynamic priority ordering of the requirements, i.e., one that depends on the path through the tree of strategies taken. This is because the $\mathcal{P}_x$-strategies again cannot choose the number they will enumerate large as usual but must enumerate their fixed number x.

So fix an arbitrary effective priority ordering of the $\mathcal{N}$- and $\mathcal{M}$-requirements only. (For simplicity, assume that in this ordering, the highest-priority requirement is an $\mathcal{N}$-requirement.) We inductively define a tree $T \subseteq \Lambda^{<\omega}$ and the assignment of requirements to all strategies $\alpha \in T$. We start by assigning to the empty node $\emptyset \in T$ the highest-priority $\mathcal{N}$-requirement and letting its immediate successors be $\langle s \rangle$ for $s \in \omega^*$. Assume that we have determined that $\alpha \in T$ and have assigned a requirement to $\alpha^- = \alpha\restriction(|\alpha|-1)$. If α^- is assigned an $\mathcal{N}$- or $\mathcal{M}$-requirement then we assign to α the requirement $\mathcal{P}_x$ for the least x such that $\mathcal{P}_x$ is not assigned to any $\beta \subset \alpha$. Otherwise, let s be the greatest integer such that $\beta\,\hat{}\,\langle s \rangle \subseteq \alpha$ for some $\beta \subset \alpha$. If there is some (least) $x \leq s$ such that $\mathcal{P}_x$ is not assigned to any $\beta \subset \alpha$ then we assign $\mathcal{P}_x$ to α, else we assign to α the highest-priority $\mathcal{N}$- or $\mathcal{M}$-requirement not assigned to any $\beta \subset \alpha$. (Note that in this assignment, infinitary outcomes of $\mathcal{M}$-strategies are ignored when deciding whether or not to assign another $\mathcal{P}$-strategy.)

We can again formulate a lemma about the assignment of requirements:

Lemma 1.

(i) *For any path $p \in [T]$ and any requirement, there is a strategy $\alpha \subset p$ assigned to that requirement.*

(ii) *For any strategy $\alpha \in T$ assigned to a requirement $\mathcal{P}_x$, say, there is at most one $\mathcal{N}$- or $\mathcal{M}$-strategy $\beta \subset \alpha$ with $\beta\hat{\ }\langle s\rangle \subseteq \alpha$ for some $s \geq x$. (This will be the longest $\mathcal{N}$- or $\mathcal{M}$-strategy $\beta \subset \alpha$ with $\beta\hat{\ }\langle s\rangle \subseteq \alpha$ for some $s \in \omega$.)*

Proof. Same as for the corresponding lemma for the Sacks Splitting Theorem. (However, we can no longer choose a function i_α since the $\mathcal{M}$-strategies do not favor one of the A's.)

Construction: During the course of stage s, let each $\mathcal{M}$-strategy $\alpha \in T$ enumerate $\vartheta(x)$ into B at any time as soon as $\Theta(B;x)$ is currently defined but equals neither $\Psi(A_0 \oplus B;x)$ nor $\Psi(A_1 \oplus B;x)$ (with computations that have already existed at some α-expansionary stage). Then initialize all $\xi \geq \beta\hat{\ }\langle s\rangle'$ (for the current (finitary) outcome s' of α) and proceed.

At each substage t of stage $s \geq t$, let a strategy α of length t be eligible to act iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$, and let α act according to its description above. If α is a $\mathcal{P}_x$-strategy ready to enumerate x (which is currently not in $A_0 \cup A_1$) into A_j for some $j \leq 1$, then fix the highest-priority $\beta\hat{\ }\langle s\rangle' \leq \alpha$ where β is an $\mathcal{N}$- or $\mathcal{M}$-strategy with current (finitary) outcome $s' \geq x$. If β is an $\mathcal{N}^i_\Phi$-strategy then α enumerates x into A_{1-i}. If β is an $\mathcal{M}$-strategy then fix $i \leq 1$ such $\Psi(A_i \oplus B)\restriction(x+1) = \Theta(B)\restriction(x+1)$ for maximal x (with the same computations as at the last α-expansionary stage) and let α enumerate x into A_{1-i}. In either case, initialize all strategies $\geq \beta\hat{\ }\langle s'\rangle$ before proceeding to the next substage. (If β fails to exist then simply let α enumerate x into A_0.)

At the end of stage s, (re)define $\Gamma(W;x) = B(x)$ with use $\gamma(x) = x$ for all $x \leq s$ for which $\Gamma(W;x)$ is currently undefined.

Remark. Note the priority ordering of the nodes β entering into the decision of whether to enumerate a number x into A_0 or A_1: We give an $\mathcal{M}$-strategy β lower priority in this decision than any $\mathcal{N}$-strategy $\beta' \supseteq \beta\hat{\ }\langle\infty\rangle$ since β does not really have a preference as it can correct its Θ.

Verification: Clearly, requirement $\mathcal{R}$ is met.

Denote by $f \in [T] \cup T$ the true path of the construction, i.e., the lim inf of f_s as s tends to infinity. Recall that f may be of finite length, i.e., be an $\mathcal{N}$-strategy on the tree with infinitary outcome. In that case, we only need to verify the satisfaction of the requirements handled by strategies $\subseteq f$.

Lemma 2. *Every strategy $\alpha \subseteq f$ is initialized at most finitely often.*

Proof. For the sake of a contradiction, fix the shortest $\alpha \subseteq f$ which is initialized infinitely often. Then this must be caused by the initialization of strategies $\geq \beta\hat{\ }\langle s\rangle$ for $\mathcal{N}$- or $\mathcal{M}$-strategies β with current (finitary) outcome s at times when a $\mathcal{P}$-strategy enumerates into A_0 or A_1, or when an $\mathcal{M}$-strategy corrects Θ. Then, since $\alpha \subseteq f$, there is a fixed bound on s not depending on β, so α can be initialized at most finitely often.

Lemma 3. *Fix a stage s and an $\mathcal{N}^i$- or $\mathcal{M}$-strategy α with current finitary outcome s' at stage s. If α is an $\mathcal{M}$-strategy then fix $i \leq 1$ such $\Psi(A_i \oplus B)\restriction(x+1) = \Theta(B)\restriction(x+1)$ for maximal x (with the same computations as at the previous α-expansionary stage). Then for any $\mathcal{M}$-strategy β with $\beta\hat{\ }\langle\infty\rangle \subseteq \alpha$ and for any x*

with $\vartheta_\beta(x) \leq s'$*, we have* $\psi_\beta(A_i \oplus B; x) \leq s'$ *(and so, in particular,* $\Psi_\beta(A_i \oplus B; x)$ *is defined). Furthermore,* $(A_i \oplus B)\lceil(s'+1)$ *has not changed between stages* s' *and* s*.*

Remark. This lemma thus states that for each such α, s' is an $A_i \oplus B$-configuration (in the sense of Slaman).

Proof. Suppose the first claim of the lemma first fails at a stage s. Then at stage s, $\psi_\beta(A_i \oplus B; x)$ increases due to A_i- or B-enumeration but α is not initialized at the same time.

First suppose that this is due to A_i-enumeration of a number x by a $\mathcal{P}$-strategy β'. But then α is one of the strategies deciding whether x should enter A_0 or A_1, and since α is overruled, it is also initialized unless α is an $\mathcal{M}$-strategy overruled by a strategy $\supseteq \alpha\hat{}\langle\infty\rangle$. But in that case, α immediately corrects $\Theta(B; x)$ since then $\Psi_\beta(A_{1-i} \oplus B; x)$ was destroyed before and after the last α-expansionary stage.

On the other hand, suppose that $\Psi_\beta(A_i \oplus B; x)$ is destroyed by an $\mathcal{M}$-strategy β'. Since β is injured but not initialized we must have $\beta'\hat{}\langle\infty\rangle \subseteq \beta$. But then β' cannot correct $\Theta_{\beta'}$ by the inductive hypothesis.

The second half of the lemma follows by the same argument.

Lemma 4. *Requirement* $\mathcal{G}$ *is satisfied.*

Proof. By the action at the end of each stage and the fact that the use $\gamma(z)$ is always set to z, $\Gamma(W)$ is clearly total.

So suppose that $\Gamma(W; z) \neq B(z)$. Then $\Gamma(W; z)$ was permanently defined to be 0 at some stage s, and at some stage $s_2 > s$, z entered B through the enumeration of $z = \vartheta(x)$ by some $\mathcal{M}_\Psi$-strategy α.

Now suppose that α last defined $\Theta(B; x)$ at a stage $s' < s_2$. Then at stage $s_1 \in [s', s_2]$ and at stage s_2 itself, the computations $\Psi(A_0 \oplus B; x)$ and $\Psi(A_1 \oplus B; x)$ are destroyed due to the enumeration by $\mathcal{P}$-strategies β_1 and β_2, respectively, either directly or via the subsequent B-enumeration of other $\mathcal{M}$-strategies. Let s_0 be the greatest α-expansionary stage $\leq s_1$. Then there are no α-expansionary stages in the interval $(s_0, s_2]$, and so Lemma 3 applies at stage s_2 with current outcome s_0 for α. So the injury to α at stage s_2 must be due to enumeration by a $\mathcal{P}$-strategy $\beta > \alpha$ or by an $\mathcal{M}$-strategy $\beta \supseteq \alpha\hat{}\langle\infty\rangle$.

If β is a $\mathcal{P}$-strategy then since α is injured at stage s_2, it must be overruled by an $\mathcal{M}$- or $\mathcal{N}$-strategy β' with $\alpha\hat{}\langle\infty\rangle \subseteq \beta' < \beta$. Since there are no α-expansionary stages in the interval $(s_0, s_2]$, β' was not initialized at stage s_1 and imposed the same restraint at stages s_1 and s_2. Thus at stage s_1, a number above the restraint of β' as enumerated, but at stage s_2, a number $x_2 \leq$ the restraint of β' is enumerated. By Lemma 3 applied to β', the use $\vartheta(x)$ is greater than the restraint of β', so $\vartheta(x) > x_2$ as desired.

On the other hand, if β is an $\mathcal{M}$-strategy then a similar argument applies, using the $\mathcal{P}$-strategy which causes β to enumerate.

Lemma 5. *Every strategy* $\alpha \subseteq f$ *ensures the satisfaction of its requirement.*

Proof. Fix s_0 least such that $\alpha \subseteq f_{s_0}$ (and so $\alpha \subseteq f_s$ for all $s \geq s_0$ by the Π_2-Lemma).

We now distinguish cases for α.

Case 1: α is a $\mathcal{P}_x$-strategy: Then α enumerates x into $A_0 \cup A_1$ iff $x \in W$.

Case 2: α is an $\mathcal{M}_\Psi$-strategy: If $\Psi(A_0 \oplus B) = \Psi(A_1 \oplus B)$ then α will make $\Theta(B)$ total (since the uses of Θ are bounded by the greater of the two Ψ-uses) and correct (by immediate Θ-correction).

Case 3: α is an $\mathcal{N}^i_\Phi$-strategy: Then α starts defining its function Δ at stage s_0. Let s be the greatest integer such that $\beta\hat{}\langle s\rangle \subseteq \alpha$ for some $\beta \subset \alpha$, or -1 if no such β exists. Note that by the way we assigned requirements and the way we initialize, no number $\leq s$ enters $A_0 \cup A_1$ after stage s_0.

If α eventually stops at Step 2 for some n then clearly $W \neq \Phi(A_i \oplus B)$. Otherwise, α will make Δ total. If we can show that no computation $\Phi(A_i \oplus B; n)$ can be destroyed once α defines $\Delta(n)$ then we can argue as in the theorem on avoiding upper cones that Δ must correctly compute W.

Suppose some strategy β enumerates a number x into A_i or B, destroying a computation $\Phi(A_i \oplus B; n)$ once α has defined $\Delta(n)$. Then β must have been overruled by a strategy $< \beta$, resulting in β being initialized and permanently off the current true path, contradicting our hypothesis.

4. Some Related Non-Injury Constructions

4.1. General remarks. Non-injury constructions often arise when a possible priority argument fails because the statement to be shown is wrong. We will illustrate this with the example of infinite injury constructions where the conflicts between the strategies are insurmountable. Typically, in this case, the main objects to be constructed are Turing functionals with some auxiliary computably enumerable sets into which any strategy may enumerate as long the functionals computing it are kept correct. Since there is therefore very little conflict between the various strategies, we end up with a list of requirements which, if they share any enumerable set being constructed at all, merely need their strategies to work on disjoint parts of each of these sets, thus giving no injury between the action for different requirements. Often, each requirement will have subrequirements, and sometimes these subrequirements may have subrequirements of their own. In that case, the strategies for a requirement and its subrequirements have to cooperate so that the subrequirements' strategies have enough flexibility to perform their tasks, e.g., by delaying the requirement's strategy in defining a shared functional so as to allow the subrequirements' strategies to first enumerate some numbers.

4.2. Finite lattices that cannot be embedded into the computably enumerable degrees. Once the embeddability of N_5 and M_3 had been shown, many conjectured *all* finite lattices to be embeddable into the computably enumerable Turing degrees. Lerman pointed out that the techniques for N_5 and M_3 at the time were not sufficient to embed a particular eight-element lattice, S_8, and this lattice was soon shown to be non-embeddable.

Theorem 14. (Lachlan and Soare (1980)) *There is an eight-element lattice, S_8, that cannot be embedded into the computably enumerable Turing degrees. (This lattice has an element w such that the elements $\leq w$ form a copy of M_3 and the elements $\geq w$ form a copy of the diamond lattice.)*

Proof. It will be easy to check that S_8 satisfies the condition NEC below; thus the result follows by the next theorem.

A few years later, the combinatorial essence of the proof for S_8 had been extracted in a sufficient condition for non-embeddability called NEC (Non-Embeddability Condition). (This condition was long conjectured also to be necessary until

Lempp and Lerman (1995) found a counterexample, L_{20}.) We begin with a definition.

Definition. Let L be a finite lattice.

(i) x, y, and z form a *critical triple* in L if they are pairwise incomparable, $x \vee z = y \vee z$, and $x \wedge y \leq z$.

(ii) L satisfies *NEC* if there are a critical triple x, y, z as well as incomparable elements p and q such that $x \leq p \wedge q \leq x \vee z \leq q$.

Theorem 15. (Ambos-Spies and Lerman (1986)) *Any finite lattice L satisfying NEC cannot be embedded into the computably enumerable Turing degrees.*

Proof. Fix $x, y, z, p, q \in L$ via which L satisfies NEC. Set $w = x \vee z = y \vee z$ and $r = p \wedge q$. Assume for the sake of a contradiction that L is embeddable, and fix computably enumerable sets P, Q, R, W, X, Y, and Z such that their degrees are the images of the corresponding lattice elements under the assumed embedding. By the lattice relations given, we may assume that $X \subseteq R \subseteq W \subseteq Q$; $R \subseteq P$; and $Y, Z \subseteq W$ with the additional condition that every number entering a set also enters all supersets (as given above) at the same time.

Furthermore, fix Turing functionals Π, Σ_0, and Σ_1 such that $W = \Pi(R \oplus Z) = \Sigma_0(X \oplus Z) = \Sigma_1(Y \oplus Z)$, and assume that these are defined and correct for every argument x at every stage $s \geq x$.

Since, by Lachlan (1966), the infima of computably enumerable degrees taken in the computably enumerable degrees and taken in all the Turing degrees coincide, it suffices to build Δ^0_2 sets E and $F = F_\Phi$ in disproving the infimum relations. We thus build a Δ^0_2-set E and Turing functionals Γ_0 and Γ_1, meeting, for all Turing functionals Φ and Ψ, the following

Requirements:

$$\mathcal{R} : E = \Gamma_0(P) = \Gamma_1(Q),$$

$$\mathcal{R}_\Phi : E = \Phi(R) \to \exists\ \Delta^0_2\text{-set } F\ \exists \Delta_0, \Delta_1 (F = \Delta_0(X) = \Delta_1(Y)), \text{ and}$$

$$\mathcal{R}_{\Phi,\Psi} : E = \Phi(R) \wedge F = \Psi(Z) \to (Q \leq_T R) \vee (P \leq_T Q) \vee \exists \Theta (W = \Theta(Z)).$$

Here the Δ^0_2-set F and the Turing reductions Δ_0 and Δ_1 are objects built by one $\mathcal{R}_\Phi$-strategy and are used only by this $\mathcal{R}_\Phi$-strategy and the $\mathcal{R}_{\Phi,\Psi}$-substrategies (with the same Φ). The Turing functional Θ is built by one $\mathcal{R}_{\Phi,\Psi}$-strategy and is only used by that strategy.

Strategy for $\mathcal{R}$: At each stage s, we (re)define $\Gamma_0(P; x) = \Gamma_1(Q; x) = E(x)$ for all $x \leq s$. Here the use is $\gamma_0(x) = \mu s' \in [x, s](P_{s'} \restriction (x+1) = P_s \restriction (x+1))$; and $\gamma_1(x) = \pi\sigma\tilde{\gamma}_1(x)$ where π is the use function of Π; $\sigma(x) = \max\{\sigma_0(x), \sigma_1(x)\}$ for the use functions σ_0 and σ_1 of Σ_0 and Σ_1, respectively; and $\tilde{\gamma}_1(x) = \mu s' \in [x, s](Q_{s'} \restriction (x+1) = Q_s \restriction (x+1))$. (The functions γ_0 and $\tilde{\gamma}_1$ are often called the *standard markers* of P and Q (with respect to the given enumeration). Their limits satisfy $\gamma_0 \equiv_T P$ and $\tilde{\gamma}_1 \equiv_T Q$. It is easy to see that γ_0 and $\tilde{\gamma}_1$ are use functions with respect to the oracles P and Q.)

We could use the permitting strategy introduced in section 2 at the example of the Friedberg-Muchnik Theorem; however, to avoid an extra two layers of permitting strategies, we choose the use of standard markers and first briefly state the following

Lemma 1. *Let $C \subseteq \omega$ be an infinite computable set.*

(i) *If $\Phi(R)$ is total then for infinitely many $x \in C$, $\tilde{\gamma}_1(x) > \varphi(x)$.*
(ii) *If $\Phi(R)$ is total then for infinitely many $x \in C$, $\gamma_1(x) > \pi\sigma\varphi(x)$.*
(iii) *If $\Phi(R)$ is total then for infinitely many $x \in C$, $\gamma_0(x), \gamma_1(x) > \pi\sigma\varphi(x)$.*

Proof. If $\Phi(R)$ is total then $\varphi \leq_T R$. For the proof of each part of the lemma, assume that the inequality fails for almost all $x \in C$. Then for (i), we can conclude $Q \equiv_T \tilde{\gamma}_1 \leq_T \varphi \leq_T R$, contradicting $Q \not\leq_T R$. (ii) is an easy consequence of (i) by composing the use functions in (i) with $\pi \circ \sigma$. For (iii), we can infer that $P \equiv_T \gamma_0 \leq_T \tilde{\gamma}_1 \oplus \pi \oplus \sigma \oplus \varphi \leq_T Q$, contradicting $P \not\leq_T Q$.

Strategy for $\mathcal{R}_\Phi$: At each stage s, we (re)define $\Delta_0(X; y) = \Delta_1(Y; y) = F(y)$ for all $y \leq s$ with use $\delta_0(y) = \delta_1(y) = y$ if undefined unless some $\mathcal{R}_{\Phi,\Psi}$-strategy requests that this definition be delayed (in which case $\Phi(R; x) \downarrow\neq E(x)$ for some x).

Strategy for $\mathcal{R}_{\Phi,\Psi}$: We fix an infinite computable set C reserved solely for this $\mathcal{R}_{\Phi,\Psi}$-strategy and proceed in substrategies n (for $n \in \omega$), letting substrategy 0 start first. Each substrategy n proceeds as follows:

1. Wait for (minimal) $x_n, y_n \in C$ greater than all current parameters of substrategies $< n$ such that $\Phi(R; x_n) = E(x_n)$; $\Psi(Z; y_n) = F(y_n)$; and $y_n, \gamma_0(x_n), \gamma_1(x_n) \geq \pi\sigma\varphi(x_n)$.
2. Set $\Theta(Z; z) = W(z)$ with use $\vartheta(z) = \psi(y_n)$ for all $z \leq \sigma\varphi(x_n)$ for which $\Theta(Z; z)$ is currently undefined.
3. Start substrategy $n + 1$.
4. From now on, if $Z\restriction(\psi(y_n)+1)$ changes at any future stage then immediately cancel all substrategies $> n$ and return to Step 1.
5. Wait for $R\restriction(\pi\sigma\varphi(x_n) + 1)$ to change. (Note here that because of $W = \Pi(R \oplus Z)$, $W\restriction(\sigma\varphi(x_n) + 1)$ cannot change unless one of the conditions of Step 4 or Step 5 applies.)
6. Change the current value of $E(x_n)$ (from 0 to 1 or vice versa). Stop the action of the corresponding $\mathcal{R}_\Phi$-strategy and of all $\mathcal{R}_{\Phi,\Psi'}$-strategies (including the further definition of the functionals Δ_0 and Δ_1, since currently $E(x_n) \neq \Phi(R; x_n)$) until reaching Step 8 or invoking Step 4.
7. Wait for $W\restriction(\varphi(x_n)+1)$ to change. (Note here that because of $W = \Sigma_0(X \oplus Z) = \Sigma_1(Y \oplus Z)$, when the condition of Step 7 applies but the condition of Step 4 fails, then both $X\restriction(y_n + 1)$ and $Y\restriction(y_n + 1)$ must have changed since the stage at which we passed to Step 6, and thus since $\Delta_0(X; y_n)$ and $\Delta_1(Y; y_n)$ were last allowed to be redefined.)
8. Change the current value of $F(y_n)$ (from 0 to 1 or vice versa) and stop (except that Step 4 may still be invoked).

Outcomes of the $\mathcal{R}_{\Phi,\Psi}$-strategy:

(1): Wait at Step 1 forever for some n: Then by Lemma 1, this is either a finitary win for $\mathcal{R}_{\Phi,\Psi}$, or else P and Q are not Turing incomparable.
(7): Wait at Step 7 forever for some n: Then $E(x_n) \neq \Phi(R; x_n)$, a finitary win for $\mathcal{R}_\Phi$ (and so the Δ's need not be total).
(8): Stop at Step 8 forever for some n: Then $F(y_n) \neq \Psi(Z; y_n)$, a finitary win for $\mathcal{R}_{\Phi,\Psi}$.
(5): Wait at Step 5 forever for each n: Then $W = \Theta(Z)$, an infinitary win for $\mathcal{R}_{\Phi,\Psi}$.

(4/1): Proceed from Step 4 to Step 1 infinitely often for some (least) n: Then Φ or Ψ is partial, an infinitary win for $\mathcal{R}_{\Phi,\Psi}$.

Construction: The construction does not require a tree of strategies since the strategies for different Φ do not interact at all. The only interaction between strategies at all lies in the fact that an $\mathcal{R}_{\Phi,\Psi}$-strategy may stop the action of the corresponding $\mathcal{R}_\Phi$-strategy and $\mathcal{R}_{\Phi,\Psi'}$-strategies. So, for each pair of Turing functionals Φ and Ψ, we fix an infinite computable set $C = C_{\Phi,\Psi}$ such that any two such sets are disjoint from each other. Now, at a stage s, the first s many strategies (under some fixed effective ordering) proceed as described above such that all $\mathcal{R}_{\Phi,\Psi}$-strategies act before the corresponding $\mathcal{R}_\Phi$-strategy, and all $\mathcal{R}_\Phi$-strategies act before the $\mathcal{R}$-strategy (so as to allow the former to change E and the F's).

Verification:

Lemma 2. *Requirement $\mathcal{R}$ is met, i.e., E is a Δ^0_2-set, $\Gamma_0(P)$ and $\Gamma_1(Q)$ are total, and they all agree.*

Proof. The $\mathcal{R}$-strategy redefines $\Gamma_0(P;x)$ and $\Gamma_1(Q;x)$ at any stage $s \geq x$ when these are undefined. Since the use functions $\gamma_0(x)$ and $\gamma_1(x)$ reach a limit for all x, $\Gamma_0(P)$ and $\Gamma_1(Q)$ must be total. $E(x)$ is allowed to change only when $R\restriction(\pi\sigma\varphi(x)+1)$ changes and $\gamma_0(x), \gamma_1(x) \geq \pi\sigma\varphi(x)$, i.e., when $P\restriction(\gamma_0(x)+1)$ and $Q\restriction(\gamma_1(x)+1)$ both change. Thus E is a Δ^0_2-set and agrees with both Γ's.

Lemma 3. *Each requirement $\mathcal{R}_\Phi$ is satisfied, i.e., if $E = \Phi(R)$ then F is a Δ^0_2-set, $\Delta_0(X)$ and $\Delta_1(Y)$ are total, and they all agree.*

Proof. Assume that $\Phi(R) = E$. Then for any Ψ, no substrategy of the $\mathcal{R}_{\Phi,\Psi}$-strategy can wait forever at Step 7 (since then $E(x) \neq \Phi(R;x)$ and $W\restriction(\varphi(x)+1)$ (and thus $R\restriction(\varphi(x)+1)$) no longer changes). Thus the definition of the Δ's is never permanently stopped, and since $\delta_0(y)$ and $\delta_1(y)$ both equal y, both Δ's must be total.

Now when $F(y)$ changes via Step 8 of a substrategy of an $\mathcal{R}_{\Phi,\Psi}$-strategy at a stage s_1 then $W\restriction(\varphi(x)+1)$ also changes at stage s_1, and, at a stage $s_0 \leq s_1$, $R\restriction(\pi\sigma\varphi(x)+1)$ has changed and $y \geq \pi\sigma\varphi(x)[s_0]$. Note that between stages s_0 and s_1, $\Delta_0(X;y)$ and $\Delta_1(Y;y)$ have not been redefined. Since $W\restriction(\varphi(x)+1)$ changes at stage s_1 but $Z\restriction(\sigma\varphi(x)+1)$ does not, both $X\restriction(\sigma\varphi(x)+1)$ and $Y\restriction(\sigma\varphi(x)+1)$ must change at stage s_1, and so both $X\restriction(y+1)$ and $Y\restriction(y+1)$ must change between stages s_0 and s_1. (Note that X and Y may change several times between stage s_0 and s_1, and the last change need not be at a number $\leq y$.) This establishes that F must be Δ^0_2 and agree with both Δ's.

Lemma 4. *Each requirement $\mathcal{R}_{\Phi,\Psi}$ is satisfied, i.e., if $E = \Phi(R)$ and $F = \Psi(Z)$ while $P \not\leq_T Q$ and $Q \not\leq_T R$ then $W = \Theta(Z)$.*

Proof. By Lemma 1, the hypotheses of Lemma 4, and the minimality of x_n and y_n, no substrategy of the $\mathcal{R}_{\Phi,\Psi}$-strategy can permanently wait at Steps 1 or 7, stop at Step 8, or return infinitely often to Step 1. Therefore all such substrategies must eventually wait at Step 5 at a stage after which $(Z \oplus R)\restriction(\pi\sigma\varphi(x)+1)$ no longer changes, which then implies that $W\restriction(\sigma\varphi(x)+1)$ no longer changes and $\Theta(Z)$ correctly computes W, contradicting $W \not\leq_T Z$.

The above lemmas imply that L is not embeddable into the computably enumerable degrees.

4.3. The Lachlan Non-Diamond Theorem. Lachlan's Non-Diamond Theorem uses a non-embeddability argument similar to the NEC argument in the previous section. It is the first priority argument in which non-uniformity plays a major role. (We present a version not using the Kleene Fixed-Point Theorem.)

Theorem 16. (Lachlan Non-Diamond Theorem (1972)) *There is no embedding of the diamond lattice into the computably enumerable Turing degrees which preserves* 0 *and* 1 *(i. e., which maps* 0 *to* $\mathbf{0}$ *and* 1 *to* $\mathbf{0}'$*).*

Proof. For the sake of a contradiction, assume such an embedding exists, and fix computably enumerable sets W_0 and W_1 such that their degrees are the images of the atoms of the diamond lattice under the assumed embedding. We build a computably enumerable set C (which is "our version" of the complete set K), meeting for all Turing functionals Ω, Φ, and Ψ, the following

Requirements:

$$\mathcal{R}_\Omega : C = \Omega(W_0 \oplus W_1) \to \exists A, \Gamma_0, \Gamma_1 (A = \Gamma_0(W_0) = \Gamma_1(W_1)),$$
$$\mathcal{R}_{\Omega,\Phi} : C = \Omega(W_0 \oplus W_1) \wedge (A = \Phi) \to \exists \Lambda_0 (W_0 = \Lambda_0) \vee \exists B, \Delta_0, \Delta_1 (B = \Delta_0(W_0) = \Delta_1(W_1)), \text{ and}$$
$$\mathcal{R}_{\Omega,\Phi,\Psi} : C = \Omega(W_0 \oplus W_1) \wedge (A = \Phi) \wedge (B = \Psi) \to (W_0 = \Lambda_0) \vee \exists \Lambda_1 (W_1 = \Lambda_1).$$

Thus our plan of attack is the following: We first build a computably enumerable set C and challenge the opponent to compute C from $W_0 \oplus W_1$ (which he claims to be complete) via a functional Ω. Based on Ω, we build a computably enumerable set $A = A_\Omega \leq_T W_0, W_1$ and challenge the opponent to compute A via a computable function Φ. Based on Ω and Φ, we build another computably enumerable set $B = B_{\Omega,\Phi} \leq_T W_0, W_1$ and again challenge the opponent to compute B via a computable function Ψ. If so, we will compute W_0 or W_1 via a computable function Λ_0 or Λ_1, respectively. (Thus we build two potential counterexamples to $\deg(W_0)$ and $\deg(W_1)$ forming a minimal pair, namely $\deg(A)$ and $\deg(B)$. It can be shown that this non-uniformity is necessary.)

For the above, the set C is global to all strategies. Each $\mathcal{R}_\Omega$-strategy builds a set A and functionals Γ_0 and Γ_1, which will be shared with all $\mathcal{R}_{\Omega,\Phi}$-strategies and $\mathcal{R}_{\Omega,\Phi,\Psi}$-strategies (for the same Ω). Each $\mathcal{R}_{\Omega,\Phi}$-strategy builds a set B and reductions Δ_0, Δ_1, and Λ_0, which will be shared with all $\mathcal{R}_{\Omega,\Phi,\Psi}$-strategies (for the same Ω and Φ). Finally, each $\mathcal{R}_{\Omega,\Phi,\Psi}$-strategy builds a reduction Λ_1 of its own.

Strategy for $\mathcal{R}_\Omega$: At each stage s, we (re)define $\Gamma_0(W_0; x) = \Gamma_1(W_1; x) = A(x)$ for all $x \leq s$ with use $\gamma_0(x) = \gamma_1(x) = x$ unless some $\mathcal{R}_{\Omega,\Phi}$-strategy requests otherwise (in which case $\Omega(W_0 \oplus W_1; c) \downarrow \neq C(c)$ for some c).

Strategy for $\mathcal{R}_{\Omega,\Phi}$: At each stage s, we (re)define $\Delta_0(W_0; y) = \Delta_1(W_1; y) = B(y)$ for all $y \leq s$ with use $\delta_0(y) = \delta_1(y) = y$ unless some $\mathcal{R}_{\Omega,\Phi,\Psi}$-strategy requests otherwise (in which case $\Phi(x) \neq A(x)$ for some x or the $\mathcal{R}_{\Omega,\Phi}$-strategy starts defining $\Lambda_0 = W_0$).

Strategy for $\mathcal{R}_{\Omega,\Phi,\Psi}$: We proceed as follows:

1. Pick c (targeted for C) larger than any number mentioned so far in the construction and keep c out of C.
2. Wait for $\Omega(W_0 \oplus W_1; c) = 0$.
3. From now on restart at Step 2 whenever $\Omega(W_0 \oplus W_1; c)$ becomes undefined before reaching Step 19, canceling all requests to other strategies as well as all Λ's, x's, and y's, and return to Step 2.

4. Set $n = 0$.
5. Pick y_n (targeted for B) larger than any number mentioned so far in the construction (so in particular $> n, \omega(c)$) and keep y_n out of B.
6. Wait for $\Psi(y_n) = 0$.
7. Set $\Lambda_1(n) = W_1(n)$.
8. Restart at Step 5 with $n+1$ (and a new y_{n+1}) while waiting at Step 9.
9. Wait for n to enter W_1.
10. Cancel all action for $n' \neq n$, discard Λ_1, and delay the redefinition of $\Delta_1(W_1; y_n)$ as well as the action of all other $\mathcal{R}_{\Omega,\Phi,\Psi'}$-strategies.

Now let the $\mathcal{R}_{\Omega,\Phi}$-strategy act as follows:

11. With the same c, still restart (as stated in Step 3) the $\mathcal{R}_{\Omega,\Phi,\Psi}$-strategy at Step 2 whenever $\Omega(W_0 \oplus W_1; c)$ becomes undefined before reaching Step 19.
12. Set $m = 0$.
13. Pick x_m (targeted for A) larger than any number mentioned in the construction so far (so in particular $> m, \omega(c)$) and keep x_m out of A.
14. Wait for $\Phi(x_m) = 0$.
15. Set $\Lambda_0(m) = W_0(m)$.
16. Restart at Step 13 with $m+1$ (and a new x_{m+1}) while waiting at Step 17.
17. Wait for m to enter W_0.
18. Cancel all action for $m' \neq m$, discard Λ_0, and delay the redefinition of $\Gamma_0(W_0; x_m)$ as well as the action of all other $\mathcal{R}_{\Omega,\Phi'}$- and $\mathcal{R}_{\Omega,\Phi',\Psi'}$-strategies.

Now let the $\mathcal{R}_{\Omega}$-strategy act as follows:

19. Enumerate c into C.
20. Wait for $W_0 \upharpoonright (\omega(c)+1)$ or $W_1 \upharpoonright (\omega(c)+1)$ to change.
21. If $W_0 \upharpoonright (\omega(c)+1)$ has changed then enumerate y_n into B, restart the definition of Γ_0 and Δ_1 as well as all action of the $\mathcal{R}_{\Omega,\Phi'}$- and $\mathcal{R}_{\Omega,\Phi',\Psi'}$-strategies, and stop the $\mathcal{R}_{\Omega,\Phi,\Psi}$-strategy forever.
22. If $W_1 \upharpoonright (\omega(c)+1)$ has changed then enumerate x_m into A, restart the definition of Γ_0 as well as all action of the $\mathcal{R}_{\Omega,\Phi'}$- and $\mathcal{R}_{\Omega,\Phi',\Psi'}$-strategies, and stop the $\mathcal{R}_{\Omega,\Phi}$- and $\mathcal{R}_{\Omega,\Phi,\Psi'}$-strategies forever.

Outcomes of the $\mathcal{R}_{\Omega,\Phi,\Psi}$-strategy:

(2): Wait at Step 2 forever for some n: Then $C(c) \neq \Omega(W_0 \oplus W_1; c)$, a finitary win for $\mathcal{R}_{\Omega}$.

(6): Wait at Step 6 forever for some n: Then $B(y_n) \neq \Psi(y_n)$, a finitary win for $\mathcal{R}_{\Omega,\Phi,\Psi}$.

(9): Wait at Step 9 eventually for all n: Then $\Lambda_1 = W_1$, an infinitary win for $\mathcal{R}_{\Omega,\Phi,\Psi}$.

(14): Wait at Step 14 forever for some m: Then $A(x_m) \neq \Phi(x_m)$, a finitary win for $\mathcal{R}_{\Omega,\Phi}$. (Note that Δ_1 is then not needed, so its redefinition can be delayed indefinitely.)

(17): Wait at Step 17 eventually for all m: Then $\Lambda_0 = W_0$, an infinitary win for $\mathcal{R}_{\Omega,\Phi}$. (Note again that Δ_1 is then not needed, so its redefinition can be delayed indefinitely.)

(20): Wait at Step 20 forever: Then $C(c) = 1 \neq 0 = \Omega(W_0 \oplus W_1; c)\downarrow$, a finitary win for $\mathcal{R}_{\Omega}$. (Note that Δ_1 and Γ_0 are then not needed, so their redefinition can be delayed indefinitely.)

(21): Stop at Step 21: Then $B(y_n) = 1 \neq 0 = \Psi(y_n)$, a finitary win for $\mathcal{R}_{\Omega,\Phi,\Psi}$.

(22): Stop at Step 22: Then $A(x_m) = 1 \neq 0 = \Phi(x_m)$, a finitary win for $\mathcal{R}_{\Omega,\Phi}$. (Note that Δ_1 is then not needed, so its redefinition can be delayed indefinitely.)

(to 2): Return to Step 2 infinitely often via Step 3: Then $C(c) \neq \Omega(W_0 \oplus W_1; c)\uparrow$, an infinitary win for $\mathcal{R}_\Omega$.

Construction: The construction does not require a tree of strategies since the strategies for different Ω do not interact at all. The only interaction between strategies at all lies in the fact that an $\mathcal{R}_{\Omega,\Phi,\Psi}$-strategy may alter the action of the corresponding $\mathcal{R}_{\Omega,\Phi}$- and $\mathcal{R}_\Omega$-strategies.

So, for each triple of Turing functionals Ω, Φ, and Ψ, we fix an infinite computable set $C = C_{\Omega,\Phi,\Psi}$ such that any two are disjoint from each other. Now, at a stage s, the first s many strategies (under some fixed effective ordering) proceed as described above, except that each strategy may pick the parameters c, y_n, and x_m only from its own set C.

Verification:

Lemma 1. *Requirement $\mathcal{R}_\Omega$ is satisfied.*

Proof. Given x, the $\mathcal{R}_\Omega$-strategy tries to redefine $\Gamma_i(W_i; x)$ (for $i \leq 1$) at any stage $s \geq x$ at which it is not delayed by an $\mathcal{R}_{\Omega,\Phi}$-strategy (in which case $C(c) \neq \Omega(W_0 \oplus W_1; c)\downarrow$ for some c). Since the uses $\gamma_i(x)$ are constant, $\mathcal{R}_\Omega$ is met.

Lemma 2. *Each requirement $\mathcal{R}_{\Omega,\Phi}$ is satisfied.*

Proof. We distinguish three cases:

Case 1: The redefinition of Δ_1 by, or the action of, the $\mathcal{R}_{\Omega,\Phi}$-strategy is never delayed permanently: Then, since the $\mathcal{R}_{\Omega,\Phi}$-strategy can act infinitely often, we see as in the proof of Lemma 1 that $B = \Delta_0(W_0) = \Delta_1(W_1)$ if the hypotheses of $\mathcal{R}_\Omega$ apply.

Case 2: The $\mathcal{R}_{\Omega,\Phi}$-strategy is delayed permanently by some $\mathcal{R}_{\Omega,\Phi'}$-strategy: Then $C(c) \neq \Omega(W_0 \oplus W_1; c)\downarrow$ for some c.

Case 3: Some $\mathcal{R}_{\Omega,\Phi}$-strategy waits at Step 17 eventually forever for all m: Then $\Lambda_0 = W_0$.

Lemma 3. *Each requirement $\mathcal{R}_{\Omega,\Phi,\Psi}$ is satisfied.*

Proof. We distinguish two cases:

Case 1: The action of the $\mathcal{R}_{\Omega,\Phi}$-strategy is never delayed permanently: Then, since the $\mathcal{R}_{\Omega,\Phi\Psi}$-strategy can act infinitely often, we see as in the proof of Lemma 2 that $\Lambda_1 = W_1$ unless the strategy waits permanently at Steps 2 or 6 for some fixed n.

Case 2: The $\mathcal{R}_{\Omega,\Phi,\Psi}$-strategy is delayed permanently by some $\mathcal{R}_{\Omega,\Phi'}$- or $\mathcal{R}_{\Omega,\Phi,\Psi'}$-strategy: Then $C(c) \neq \Omega(W_0 \oplus W_1; c)\downarrow$ for some c, or $\Lambda_0 = W_0$.

The above lemmas imply the theorem.

5. Degenerate $\mathbf{0}'''$- or Σ_3-Constructions

5.1. General remarks. Degenerate $\mathbf{0}'''$- or Σ_3-constructions share some of the properties of infinite injury constructions as well as some of the properties of $\mathbf{0}'''$-constructions: On the one hand, each strategy has an outcome computable by a $\mathbf{0}''$-oracle (unless a hypothesis of the theorem turns out to be wrong, e.g., some given set turns out to be computable in some other given set contrary to hypothesis). On the other hand, the conditions in the requirements are Π_3-conditions as in true $\mathbf{0}'''$- or Π_3-constructions.

All degenerate $\mathbf{0}'''$- or Σ_3-constructions share the following features:

The *requirements* are typically of the form

$$(P \to S)$$

where $P \equiv \forall n P_0(n)$ and $S \equiv \forall n S_0(n)$ are Π_3-conditions, possibly relative to some computably enumerable oracles which we are building, and where the Π_3-condition S contradicts the hypotheses of the theorem so that if P is found true then we contradict the theorem's hypotheses and the remaining requirements need not be satisfied.

A Σ_3*-strategy* is of the form

1. Pick some parameters and set $n = 0$.
2. Wait for (an instance of) $P_0(n)$ to hold.
3. Try to ensure the validity of $P_0(n)$, ensure another instance of S, and restart at Step 2 with $n + 1$.
4. From now on, check whether $P_0(n)$ still holds; when it fails, then go back to Step 2 (and cancel the action taken for $n' > n$).

The possible *outcomes* are then the *finitary outcomes* w_n for $\neg P_0(n)$ (i.e., eventually waiting at Step 2 forever for fixed n), the *infinitary outcomes* ∞_n for $\neg P_0(n)$ (i.e., going from Step 4 to Step 2 infinitely often for fixed n), and the $\mathbf{0}'''$-outcome ∞ for P (i.e., eventually waiting at Step 4 forever for all n). At each stage s, we define the *current outcome* to be ∞_n if the strategy has just returned from Step 4 back to Step 2 for this (least) n, and w_n if the strategy is currently stuck waiting at Step 4 for this (unique) n. A true finitary outcome is the current outcome at cofinitely many stages. A true infinitary outcome is the current outcome at infinitely many stages. (The $\mathbf{0}'''$-outcome ∞ cannot be determined so easily but it contradicts the theorem's hypotheses anyhow.) The set Λ of *finitary* and *infinitary* outcomes of a Σ_3-strategy will typically be a set of the form

$$\{\infty_0 <_\Lambda w_0 <_\Lambda \infty_1 <_\Lambda w_1 <_\Lambda \infty_2 <_\Lambda w_2 <_\Lambda \cdots\}.$$

The *tree of strategies* is an infinite branching subtree of $\Lambda^{<\omega}$ such that each node ("strategy") $\alpha \in T$ has as immediate successors all nodes $\alpha\hat{\ }\langle o\rangle$ where o ranges over the possible outcomes of α. The requirements are assigned effectively to the strategies in such a way that along any path through T, all requirements are handled. (Note that we do not put the $\mathbf{0}'''$-outcome of Σ_3-strategies on the tree of strategies since under this outcome, the theorem's hypotheses are violated, and so we need not deal with further requirements.)

The *construction* is performed in ω many stages s. Each stage consists of substages $t \leq s$. At substage t, the strategy α of length t eligible to act is chosen such that for all $\beta \subset \alpha$, $\beta\hat{\ }\langle o\rangle \subseteq \alpha$ iff β currently has current outcome o. Strategy α will then proceed according to its strategy from where it left off the last time it was eligible to act. We define the *current true path* f_s at stage s to be the longest strategy eligible to act at stage s. We will also use the feature of *initialization* here for the first time, namely, any strategy $\alpha > f_s$ will be *initialized* at the end of stage s by making α start all over and making all its parameters, functionals, etc., undefined.

For the *verification*, define the *true path* f to be the lim inf of the current true path inductively defined by

$$f(n) = \min{}_s\{o \in \Lambda \mid \exists^\infty s((f\restriction n)\hat{}\langle o\rangle \subseteq f_s)\}.$$

(This lim inf will only exist if no strategy along the true path has the $\mathbf{0}'''$-outcome. In that case, we let the true path be a node of the tree, namely, the longest node such that we can define this lim inf; so this node will have the $\mathbf{0}'''$-outcome.) To inductively verify the satisfaction of a requirement $\mathcal{R}$, consider the $\mathcal{R}$-strategy $\alpha \subseteq f$. Strategy α will be eligible to act at some (least) stage s_0 such that after stage s_0, no strategy $\beta <_L \alpha$ is eligible to act. From then on, α will be eligible to act at infinitely many stages. Since no strategy $\beta <_L \alpha$ will be eligible to act after stage s_0, since α has a correct guess about all strategies $\beta \subset \alpha$, and since α has higher priority than all strategies $\beta > \alpha$, α will not be injured after stage s_0. Thus α will either have a finitary or infinitary outcome w_n or ∞_n, allowing f to have length greater than $|\alpha|$; or α will have the $\mathbf{0}'''$-outcome, contradicting the theorem's hypotheses.

5.2. The (Strong) Thickness Lemma. The Thickness Lemma (of which we present below the so-called "strong version") was the first infinite injury argument. Ironically, the strong version is actually a Σ_3-argument, i.e., combinatorially a bit more complex than a "regular" infinite injury or Π_2-argument. The Thickness Lemma (and its variations) can be used to prove a number of related results, such as the Sacks Jump Inversion Theorem presented in a previous chapter.

Theorem 17. (Shoenfield (1961) for $W = \emptyset'$) *Given computably enumerable sets V and W such that $V^{[<e]} \not\geq_T W$ for all $e \in \omega$, there is a computably enumerable set $A \subseteq V$ such that $A \not\geq_T W$ and $A^{[e]} =^* V^{[e]}$ for all $e \in \omega$.*

(Here $X^{[e]} = \{\langle x, e, |\rangle\langle x, e, \in\rangle X\}$, and similarly for $X^{[<e]}$ and $X^{[\geq e]}$; also, $X =^* Y$ if their symmetric difference is finite.)

Proof. We build a computably enumerable set A, meeting, for all $e \in \omega$ and all Turing functionals Φ the following

Requirements:

$$\begin{aligned}\mathcal{R} &: A \subseteq V,\\ \mathcal{P}_e &: A^{[e]} =^* V^{[e]}, \text{ and}\\ \mathcal{N}_\Psi &: W = \Psi(A) \to \exists\Delta\, \exists e\, (W = \Delta(V^{[<e]})).\end{aligned}$$

Here Δ is a partial computable functional built by us (i.e., by one of our strategies).

Strategy for $\mathcal{R}$: This is a trivial global requirement which only allows numbers already in V to enter A.

Strategy for $\mathcal{P}_e$: This strategy merely enumerates all numbers from $V^{[e]}$ into $A^{[e]}$.

Outcomes of the $\mathcal{P}_e$-strategy: Since $V^{[e]}$ could be a very complicated set, we cannot give this strategy distinct outcomes. (So the strategy only has a default outcome 0, and strategies of lower priority will have to deal with the $\mathcal{P}_e$-strategy's actions without additional information.)

Strategy for $\mathcal{N}_\Psi$: The strategy builds a "local" partial computable function (i.e., only this strategy may make definitions for Δ whereas other $\mathcal{N}$-strategies define their own versions of Δ).

1. Set $n = 0$.
2. Wait for $\Psi(A)\restriction(n+1) = W\restriction(n+1)$.
3. Try to preserve $\Psi(A)\restriction(n+1)$ by restraining numbers from entering the set $A^{[\geq e]}\restriction(\psi(n)+1)$, set $\Delta(V^{[<e]}; n) = W(n)$ with use $\delta(n) = \psi(n)$ (for some fixed e depending only on the strategy's location on the tree), and restart at Step 2 with $n+1$.
4. From now on, check whether $V^{[<e]}\restriction(\psi(n)+1)$ has changed. If it changes without a corresponding $A\restriction(\psi(n)+1)$-change, then redefine $\Delta(V^{[<e]}; n) = W(n)$ with the same use; otherwise, cancel the action for all $n' \geq n$ and return to Step 2.

Outcomes of the $\mathcal{N}_\Psi$-strategy:

w_n: Wait at Step 2 forever for fixed n: Then $W\restriction(n+1) \neq \Psi(A)\restriction(n+1)$.

∞_n: Return from Step 4 to Step 2 infinitely often for fixed (least) n: Then $\Psi(A; n)$ is undefined. (Note here that $\Psi(A; n)$ cannot be redefined in Step 4 infinitely often without returning to Step 2.)

∞: Eventually wait at Step 4 for each n: Then $W = \Delta(V^{[<e]})$, and so $W \leq_T V^{[<e]}$ contrary to hypothesis.

We let

$$\Lambda = \{\infty_0 <_\Lambda w_0 <_\Lambda \infty_1 <_\Lambda w_1 <_\Lambda \infty_2 <_\Lambda w_2 <_\Lambda \cdots <_\Lambda 0\}$$

be the set of finitary and infinitary outcomes.

Tree of strategies: Effectively order the $\mathcal{N}$- and $\mathcal{P}$-requirements (of order type ω) such that $\mathcal{P}_e$ has higher priority than $\mathcal{P}_{e'}$ when $e < e'$. Inductively define a tree $T \subseteq \Lambda^{<\omega}$ by assigning to all strategies α of length i the ith requirement in this list and by letting $\alpha\hat{\ }\langle o\rangle$ be the immediate successors of α where o ranges over all possible (finitary and infinitary) outcomes of α.

Construction: Each $\mathcal{N}$-strategy $\alpha \in T$ will define its functional $\Delta(V^{[<e]})$ where e is least such that there is no $\mathcal{P}_e$-strategy $\subset \alpha$.

Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$. A $\mathcal{P}$-strategy α is allowed to enumerate a number x into A only if $x >$ than the current A-restraint of any strategy $\beta < \alpha$. At the end of stage s, initialize all $\mathcal{N}$-strategies $\beta > f_s$ by canceling their functionals and resetting their parameter n to 0.

Verification: Denote by $f \in [T] \cup T$ the true path of the construction, i.e., the lim inf of f_s as s tends to infinity. Recall that f may be of finite length, i.e., be a node on the tree with $\mathbf{0}'''$-outcome. (We then only need to verify the satisfaction of the requirements handled by strategies $\subseteq f$.)

Lemma. *Every strategy $\alpha \subseteq f$ ensures the satisfaction of its requirement. If $\alpha \subset f$ is an $\mathcal{N}$-strategy and α^+ is its immediate successor along f then α's A-restraint is constant at all stages at which α^+ is eligible to act after the last stage at which α^+ is initialized. If $\alpha = f$ then α contradicts the theorem's hypothesis.*

Proof. Fix $s_0 \geq |\alpha|$ least such that $\alpha \leq f_s$ for all $s \geq s_0$; so α will not be initialized after stage s_0. We now distinguish cases for α.

Case 1: α is a $\mathcal{P}$-strategy: Then α can clearly enumerate almost every number it wishes to enumerate since the strategies $\beta < \alpha$ eventually impose constant restraint at the stages at which α is eligible to act.

Case 2: α is an $\mathcal{N}$-strategy: Then α starts defining its functional $\Delta(V^{[<e]})$ at stage s_0. If α eventually stops at Step 2 for some n then clearly $W \neq \Psi(A)$. If α goes from Step 4 to Step 2 infinitely often for some fixed n then $\Psi(A;n)$ is undefined. In either of those two cases, the A-restraint will be $\psi(n-1)$ at stages at which α's current outcome is the true outcome, so once α^+ is no longer initialized, the A-restraint will be constant at stages at which α^+ is eligible to act.

Otherwise, α will make $\Delta(V^{[<e]})$ total. $\Delta(V^{[<e]})$ must correctly compute W since as in the proof of the original Friedberg-Muchnik Theorem, we can argue that a computation $\Psi(A;n)$ cannot be destroyed once found unless $A^{[<e]}\restriction(\psi(n)+1)$ and thus $V^{[<e]}\restriction(\psi(n)+1)$ changes. Namely, no strategy $\beta > \alpha$ can destroy $\Psi(A;n)$ by α's A-restraint; and no strategy $\beta <_L \alpha$ is eligible to act after stage s_0; so the only strategies that can injure $\Psi(A;n)$ are $\mathcal{P}$-strategies $\beta \subset \alpha$. But note that if $s_1 < s_2$ are two consecutive stages $\geq s_0$ at which α is eligible to act, then any number x entering $V^{[<e]}$ at a stage $\in (s_1, s_2]$ must either be enumerated by α into A by stage s_2, or x else will never enter A since after stage s_0, the A-restraint of all strategies $\beta' < \alpha$ (which could potentially restrain β from enumerating x) is constant at all stages at which α is eligible to act.

5.3. The Sacks Density Theorem. The Sacks Density Theorem culminated the early investigations into the structure of the computably enumerable degrees, exhibiting the "nice" features of this structure.

Theorem 18. (Sacks (1964)) *Given computably enumerable degrees* $\mathbf{u} < \mathbf{v}$*, there is a computably enumerable degree* $\mathbf{a}$ *with* $\mathbf{u} < \mathbf{a} < \mathbf{v}$*. (Thus the computably enumerable Turing degrees are densely ordered.)*

Proof. Fix computably enumerable sets $U \in \mathbf{u}$ and $V \in \mathbf{v}$. We need to build a computably enumerable set A (setting $\mathbf{a} = \deg(A \oplus U)$) and a Turing functional Θ, meeting for all Turing functionals Φ and Ψ the following

Requirements:

$$\begin{aligned} &\mathcal{R} : A = \Theta(U \oplus V), \\ &\mathcal{N}_\Phi : V = \Phi(A \oplus U) \to \exists \Gamma(V = \Gamma(U)), \text{ and} \\ &\mathcal{P}_\Psi : A = \Psi(U) \to \exists \Delta(V = \Delta(U)). \end{aligned}$$

Here Γ and Δ are Turing functionals built by us (i.e., by one of our strategies).

Strategy for $\mathcal{R}$: At the end of each stage s, this global strategy (re)defines $\Theta(U \oplus V; x) = A(x)$ for each $x \leq s$ for which $\Theta(U \oplus V; x)$ is now undefined. The use will be specified later and can only be increased finitely often. No strategy may enumerate a number x into A while $\Theta(U \oplus V; x) \downarrow= 0$.

Strategy for $\mathcal{N}_\Phi$: This is the Sacks preservation strategy from the Sacks Splitting Theorem, modified to accommodate the additional injury due to U-changes. The strategy builds a "local" partial computable function (i.e., only this strategy may make definitions for Γ whereas other $\mathcal{N}$-strategies define their own versions of Γ).

1. Set $n = 0$.
2. Wait for $V\restriction(n+1) = \Phi(A \oplus U)\restriction(n+1)$.

3. Set $\Gamma(U;n) = V(n)$ with use $\gamma(n) = \varphi(n)$, try to preserve $\Phi(A \oplus U)\restriction(n+1)$ by restraining numbers from entering $A\restriction(\varphi(n)+1)$, and restart at Step 2 with $n+1$ in place of n.
4. From now on, wait for (i) $U\restriction(\varphi(n)+1)$ to change, or (ii) n to enter V.
5. If (i) applies then cancel all action for $n' > n$ and the A-restraint for n and return to Step 2.
6. If (ii) applies then stop all action for $n' > n$ and wait for (i) to apply also. When it does, proceed as in Step 5.

Outcomes of the $\mathcal{N}_\Phi$-strategy:

w_n: Wait forever at Step 2 or Step 6 for this n: Then $V\restriction(n+1) \neq \Phi(A\oplus U)\restriction(n+1)$.

∞_n: Return from Step 5 or 6 to Step 2 infinitely often for this (least) n: Then $\Phi(A \oplus U; n)$ is undefined.

∞: Eventually wait at Step 4 for each n: Then $V = \Gamma(U)$ contrary to the hypothesis of the theorem.

Current outcomes of the $\mathcal{N}$-strategy: For reasons that will become apparent when we further discuss the definition of Θ and its uses, we will collapse the outcomes ∞_n and w_n into one outcome n. We thus define the current outcome to be n if the strategy just returned from Step 5 or 6 to Step 2 (for this n), or if the strategy is stuck waiting at Step 2 or 6 (for this n). (Note that once the strategy starts up at Step 1 for $n = 0$, there will always be exactly one n for which this applies.)

Strategy for $\mathcal{P}_\Psi$: This is the Sacks coding strategy, threatening to code V into A if the opponent threatens to compute A from U. The strategy builds a "local" partial computable function (i.e., only this strategy may make definitions for Δ whereas other $\mathcal{P}$-strategies define their own versions of Δ).

1. Set $n = 0$.
2. Pick a "coding location" c_n for n (targeted for A) larger than any number mentioned so far in the construction and keep c_n out of A.
3. If currently $n \in V$ then set $\Delta(U;n) = 1$ for all possible oracles for which no previous definition applies and restart at Step 2 with $n+1$ in place of n.
4. Otherwise, wait for $A\restriction(c_n+1) = \Psi(U)\restriction(c_n+1)$.
5. Set $\Delta(U;n) = V(n)$ with use $\delta(n) = \psi(c_n)$ and restart at Step 2 with $n+1$ in place of n.
6. From now on, wait for (i) $U\restriction(\psi(c_n)+1)$ to change, or (ii) n to enter V.
7. If (i) applies then cancel all action for $n' > n$ and return to Step 3.
8. If (ii) applies then put c_n into A, stop all action for $n' > n$, and and wait for (i) to apply also. When it does, proceed as in Step 7.

Outcomes of the $\mathcal{P}$-strategy:

w_n: Wait at Step 4 or 8 forever for this n: Then $A\restriction(c_n+1) \neq \Psi(U)\restriction(c_n+1)$.

∞_n: Return from Step 7 or 8 to Step 3 infinitely often for this (least) n: Then $\Psi(U;y)$ is undefined for some $y \leq c_n$.

∞: Eventually wait at Step 6 or Step 3 for each n: Then $V = \Delta(U)$ contrary to the hypothesis of the theorem.

Current outcomes of the $\mathcal{P}$-strategy: We again need to collapse outcomes ∞_n and w_n and define the current outcome to be n if the strategy just returned from Step 7 or 8 to Step 3 (for this n), or if the strategy is stuck waiting at Step 4 or 8 (for this n). (Note that once the strategy starts up at Step 1 for $n = 0$, there will always be exactly one n for which this applies.)

Tree of strategies: We let $\Lambda = \omega$ be the set of finitary and infinitary outcomes and effectively order the $\mathcal{N}$- and $\mathcal{P}$-requirements (of order type ω). We now assign to all strategies $\alpha \in T$ of length e the eth requirement in this list. ($\mathcal{R}$ is a global requirement not represented by a strategy on the tree.)

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$. (We will show in the verification below that the $\mathcal{P}$-strategies will automatically respect $\mathcal{R}$ by not enumerating any number n into A while $\Theta(U \oplus V; n) \downarrow = 0$.) A strategy will believe a computation at stage s only if it already existed at the last stage $s' < s$ at which $\alpha \subseteq f_{s'}$; otherwise, α will assume the computation to be undefined. (So, in particular, if s is the first stage at which α is eligible to act then α will believe no computation at all.)

At the end of each substage t, if α has moved from one step to another for some n at substage t, then initialize all $\beta >_L \alpha\hat{}\langle n\rangle$ for the least such n; also initialize all $\beta \supseteq \alpha\hat{}\langle n\rangle$ if n entered V since the last substage s' at which $\alpha \subseteq f_{s'}$. (Here initializing β means canceling its functional Γ or Δ and all its coding locations c_n, and resetting its parameter n to 0.)

At the end of stage s, $\mathcal{R}$ redefines $\Theta(U \oplus V; x) = A(x)$ for all $x \leq s$ for which $\Theta(U \oplus V; x)$ is currently undefined. The use $\vartheta(x)$ is defined to be

(i) 0, if currently $x \in A$, or if x is currently not the coding location of any $\mathcal{P}$-strategy; and

(ii) n, if $x \notin A$, x is currently the coding location c_n of a $\mathcal{P}$-strategy, and $n \notin V$.

Otherwise, x is the current coding location of a $\mathcal{P}$-strategy α for which $x \notin A$ but $n \in V$ and $\alpha <_L f_s$ (since otherwise α would have been initialized at stage s or would have enumerated x into A at stage s). Fix the longest node $\beta_0 \subset \alpha, f_s$. The use $\vartheta(x)$ is then defined to be

(iii)

$$\min(\{\varphi_\beta(m) \mid \beta \text{ is an } \mathcal{N}_\Phi\text{-strategy with } \beta_0 \subset \beta\hat{}\langle m\rangle \subseteq \alpha\} \cup \\ \{\psi_\beta(c_{\beta,m}) \mid \beta \text{ is a } \mathcal{P}_\Psi\text{-strategy with } \beta_0 \subset \beta\hat{}\langle m\rangle \subseteq \alpha\}).$$

(Note here that at least the use for β_0 in the above expression must be finite since $\beta_0\hat{}\langle m\rangle <_L f_s$ for the corresponding m.)

Verification: Denote by $f \in [T] \cup T$ the true path of the construction, i.e., the lim inf of f_s as s tends to infinity. Recall that f may be of finite length, i.e., be a node on the tree with $\mathbf{0}'''$-outcome. (We then only need to verify the satisfaction of the requirements handled by strategies $\subseteq f$.) We proceed in two lemmas.

Lemma 1. *Requirement $\mathcal{R}$ is satisfied.*

Proof. Fix argument x of $\Theta(U \oplus V)$.

We first show that $\Theta(U \oplus V; x)$ is defined. Since $\mathcal{R}$ redefines $\Theta(U \oplus V; x)$ at all but finitely many stages at which it is undefined, we merely need to show that the use $\vartheta(x)$ is bounded. This is clear from the construction unless x is the permanent coding location c_n of some $\mathcal{P}$-strategy α (which is thus not initialized after picking x), $n \in V$, and $x \notin A$. So fix the stage s_0 at which n enters V. Since $\alpha <_L f$ or $f \subset \alpha$ (where, in the latter case, f is a node on the tree, so this strategy has true

Π_3-outcome ∞), fix the longest node $\beta \subset \alpha, f$ as well as β's outcome m along α. Fix the (least) stage s_1 such that $f_s >_L \beta\hat{}\langle m\rangle$ for all $s \geq s_1$. But then $\varphi_\beta(m)$ or $\psi_\beta(c_{\beta,m})$ is one of the uses in (iii) of the definition of $\Theta(U \oplus V; x)$ after stage s_1; so this use is bounded as desired.

Finally we need to show that $\Theta(U \oplus V)$ correctly computes A. Fix an argument x and assume that $\Theta(U \oplus V; x) \neq A(x)$. Then x must be enumerated as the coding location c_n by some $\mathcal{P}$-strategy α at a stage s_2 after a stage $s_0 \leq s_2$ at which n enters V. Since the coding location c_n is chosen at a stage $< s_0$, the use $\vartheta(x)$ is always chosen to be n before stage s_0, so at the end of stage s_0, $\Theta(U \oplus V; x)$ is redefined. Since by our assumption, $\Theta(U \oplus V; x) = 0$, this computation must become defined permanently before stage s_2, say at a stage $s_1 \in [s_0, s_2)$. Since x is not enumerated into A until stage s_2, and since α is not initialized between stages s_0 and s_2, $\alpha <_L f_{s_1}$ but $\alpha \subset f_{s_2}$. But all the computations $\Phi(A \oplus U; m)$ or $\Psi(U)\restriction(c_{\beta,m})$ for the β's used in the definition of $\vartheta(x)$ at stage s_1 become undefined between stages s_1 and s_2, or their corresponding m enters V. Thus $(U \oplus V)\restriction(\vartheta(x) + 1)$ changes between stages s_1 and s_2, contradicting the choice of s_1.

Remark. Note that requirement $\mathcal{R}$ is ensured by the following feature of the construction: Let

$$S = \{\langle \alpha, s\rangle \mid \forall s' \geq s(\alpha < f_{s'})\}, \text{ and}$$
$$T = \{\langle \alpha, s\rangle \mid \forall s' \geq s(\alpha \leq f_{s'})\}.$$

Then S and T are both computably enumerable in $U \oplus V$ by our collapsing of the outcomes ∞_n and w_n into one outcome n. Now $\mathcal{R}$ can correctly compute A since it can enumerate S and so predict whether a $\mathcal{P}$-strategy α that is ready to enumerate its coding location into A will ever be on the current true path again. We remark that the above also shows that the true path is d-computably enumerable in $U \oplus V$.

We now turn to the satisfaction of the $\mathcal{P}$- and $\mathcal{N}$-requirements by strategies along the true path of the construction.

Lemma 2. *Every strategy $\alpha \subseteq f$ ensures the satisfaction of its requirement. If $\alpha = f$ then α contradicts the theorem's hypothesis.*

Proof. Fix $s_0 \geq |\alpha|$ least such that $\alpha \leq f_s$ and α is not initialized at stage s for all $s \geq s_0$. We now distinguish cases for α.

Case 1: α is a $\mathcal{P}$-strategy: Then α will clearly satisfy its requirement as in the basic module of the strategy described above.

Case 2: α is an $\mathcal{N}$-strategy: Then α will try to satisfy its requirement as in the basic module of the strategy described above. The only possible complication is that α's A-restraint for preserving a computation $\Phi(A \oplus U; n)$ may be injured. By initialization, only a $\mathcal{P}$-strategy β comparable with $\alpha\hat{}\langle n\rangle$ can possibly injure that restraint. First consider a $\mathcal{P}$-strategy such that $\beta\hat{}\langle m\rangle \subseteq \alpha$ for some m. When β enumerates its coding location $c_{\beta,m'}$ for some $m' \leq m$ then α is initialized contrary to assumption on s_0. And any coding location $c_{\beta,m'}$ for $m' > m$ is chosen after α's A-restraint is imposed, so is chosen greater than that A-restraint. Now consider the case $\beta \supseteq \alpha\hat{}\langle n\rangle$. Then β is eligible to act at a time when α imposes A-restraint for preserving a computation $\Phi(A \oplus U; n)$ only if n has just entered V but $U\restriction(\varphi(n)+1)$ has not changed yet. But then β would have been initialized after n enters V, and so β cannot injure α.

5.4. The Robinson Jump Interpolation Theorem. This theorem combines the Sacks Jump Theorem and the Sacks Density Theorem. The proof does not present any new strategies but shows how the strategies of the two Sacks theorems can be combined.

Theorem 19. (R. W. Robinson (1971)) *Given two computably enumerable degrees* $\mathbf{u} < \mathbf{v}$ *and a degree* $\mathbf{s} \geq \mathbf{u}'$ *computably enumerable in* $\mathbf{v}$*, there is a computably enumerable degree* $\mathbf{a}$ *with* $\mathbf{a}' = \mathbf{s}$ *and* $\mathbf{u} < \mathbf{a} < \mathbf{v}$.

Proof. Fix computably enumerable sets $U \in \mathbf{u}$ and $V \in \mathbf{v}$ as well as a set $S \in \mathbf{s}$ and an enumeration operator W such that $S = W(V)$. We need to build a computably enumerable set A (setting $\mathbf{a} = \deg(A \oplus U)$) and Turing functionals Γ, Δ, and Θ, meeting for all Turing functionals Φ and Ψ and all $e \in \omega$ the following

Requirements:

$$\begin{aligned}
\mathcal{R} &: A = \Theta(U \oplus V), \\
\mathcal{N}_\Phi &: V = \Phi(A \oplus U) \to \exists \Gamma_\Phi (V = \Gamma_\Phi(U)), \\
\mathcal{P}_\Psi &: A = \Psi(U) \to \exists \Delta_\Psi (V = \Delta_\Psi(U)), \\
\mathcal{P}_e &: S(e) = \lim_s \Gamma(A \oplus U; e, s), \text{ and} \\
\mathcal{N}_e &: (A \oplus U)'(e) = \Delta(S \oplus U'; e).
\end{aligned}$$

Here Γ_Φ and Δ_Ψ are Turing functionals built by us (i.e., by one of our strategies).

The *strategies* for the first three requirements are essentially the same as those in the Sacks Density Theorem; however, the current outcomes are defined a bit differently: If, while in outcome n, an $\mathcal{N}_\Phi$- or $\mathcal{P}_\Psi$-strategy sees some $n' < n$ enter the set V and thus starts waiting for the corresponding U-change at Step 6 or 8, respectively, then the strategy's current outcome remains n until this U-change occurs. (This is necessary since the $\mathcal{N}_e$-strategies cannot predict V-changes.)

Strategy for $\mathcal{P}_e$: The strategy defines $\Gamma(A \oplus U; e, s)$ simultaneously for each s as follows:

1. At stage s, set $\Gamma(A \oplus U; e, s) = 0$ with use 0 (if $e \notin W_s(V_s; e)$) and $= 1$ with big use $\gamma(e, s)$ (otherwise). (If $e \in W_s(V_s; e)$ then set $w_s(e)$ to be the use of the computation that enumerates e into $W_s(V_s; e)$.)
2. If $\Gamma(A \oplus U; e, s)$ was set to 1 then wait for $V \upharpoonright (w_s(e) + 1)$ to change. (While waiting, reset $\Gamma(A \oplus U; e, s) = 1$ with the same use whenever it becomes undefined due to an $A \oplus U$-change.)
3. Enumerate $\gamma(e, s)$ into A and reset $\Gamma(A \oplus U; e, s) = 0$ with the same use. (From now on, reset $\Gamma(A \oplus U; e, s) = 0$ with the same use whenever it becomes undefined due to an $(A \oplus U)$-change.)

Outcomes of the $\mathcal{P}$-strategy:

- w: The strategy resets $\Gamma(A \oplus U; e, s) = 0$ for only finitely many s: Then $\Gamma(A \oplus U; e, s) = S(e)$ for almost all s. (This is the finitary outcome.)
- ∞: The strategy resets $\Gamma(A \oplus U; e, s) = 0$ for infinitely many s: Then $\Gamma(A \oplus U; e, s) = 0$ for all e and $e \notin S$. (This is the infinitary outcome.)

For the sake of the $\mathcal{R}$-requirement, we need the *current outcome* of the $\mathcal{P}$-strategy to be defined a bit differently than usual, namely, 1 if $e \in W(V)$ (via the same computation) at all stages since the last stage s' at which $\alpha \subseteq f_{s'}$, and 0 otherwise.

(Note that we thus have that V must change for the current outcome to move "left" from 1 to 0.)

Strategy for $\mathcal{N}_e$:

1. Set $\Delta(S \oplus U'; e) = 0$ with "appropriate" use (to be defined later).
2. Wait for a computation $\Phi_e(A \oplus U; e)$.
3. Set $\Delta(S \oplus U'; e) = 1$ with "appropriate" use and restrain $A \restriction (\varphi_e(e) + 1)$ to protect the computation $\Phi_e(A \oplus U; e)$.
4. Wait for $U \restriction (\varphi_e(e) + 1)$ to change.
5. Return to Step 1.

(We will show below that we can always redefine $\Delta(S \oplus U'; e)$ whenever needed.)

Outcomes of the $\mathcal{N}$-strategy:

w: Wait at Step 2 forever: Then $(A \oplus U)'(e) = 0 = \Delta(S \oplus U'; e)$ (a finitary outcome).

s: Wait at Step 4 forever: Then $(A \oplus U)'(e) = 1 = \Delta(S \oplus U'; e)$ (a finitary outcome).

∞: Return from Step 4 to Step 1 infinitely often: Then we will show that some definition $\Delta(S \oplus U'; e) = 0$ will be $(S \oplus U')$-correct, so $(A \oplus U)'(e) = 0 = \Delta(S \oplus U'; e)$ (an infinitary outcome).

Again for the sake of the $\mathcal{R}$-requirement, we need the *current outcome* of the $\mathcal{N}$-strategy to be defined a bit differently than usual, namely, 1 if $(A \oplus U)'(e) = 1$ at all stages since the last stage s' at which $\alpha \subseteq f_{s'}$ (with the same computation), and 0 otherwise. (Note that we thus have that U must change for the current outcome to move "left" from 1 to 0.)

Tree of strategies: We let $\Lambda = \omega$ be the set of finitary and infinitary outcomes and effectively order all $\mathcal{N}_\Phi$-, $\mathcal{P}_\Psi$-, $\mathcal{P}_e$- and $\mathcal{N}_e$-requirements (of order type ω). Inductively define a tree $T \subseteq \Lambda^{<\omega}$ by assigning to all strategies α of length i the ith requirement in this list and by letting $\alpha\hat{\ }\langle o \rangle$ be the immediate successors of α where o ranges over all possible current outcomes of α. (Requirement $\mathcal{R}$ is a global requirement which will not be assigned to any node on the tree but will be handled separately at the end of each stage.)

Construction: Let a strategy α of length t be eligible to act at substage t of stage $s \geq t$ iff α has the correct guess about the current outcomes of all $\beta \subset \alpha$. Here α believes a computation (or enumeration) at stage s only if that computation (or enumeration) has existed since the last stage s' at which $\alpha \subseteq f_{s'}$, otherwise α assumes that computation (or enumeration) to be undefined. (So, in particular, at the first stage at which α is eligible to act, it will believe no computation (or enumeration) at all.) Also, a $\mathcal{P}_e$-strategy α will only be allowed to enumerate a use $\gamma(e, s')$ if there is a stage $s_0 \leq s'$ such that $\alpha \subseteq f_{s_0}$ and α is not initialized between stages s_0 and s. (However, if α is prevented from resetting $\Gamma(A \oplus U; e, s')$ by this last clause then this does not affect the current outcome of α. We will show in the verification below that all the $\mathcal{P}$-strategies will automatically respect $\mathcal{R}$ by not enumerating any number n into A while $\Theta(U \oplus V; n) \downarrow= 0$.)

At every substage at which an $\mathcal{N}_e$-strategy α is eligible to act, it will set $\Delta(X \oplus Y; e)$ (if possible) equal to its guess about $(A \oplus U)'(e)$ for all oracles $X \oplus Y$ satisfying:

(i) $X(i) = 0$ for all $\mathcal{P}_i$-strategies β with $\beta\hat{\ }\langle 0 \rangle \subseteq \alpha$;
(ii) $X(i) = 1$ for all $\mathcal{P}_i$-strategies β with $\beta\hat{\ }\langle 1 \rangle \subseteq \alpha$;
(iii) $Y(c) = 0$ where c is a code for the Σ_1-question "Is there a future stage at which the current true path is to the left of α?";

(iv) $Y(c') = 1$ for all such codes c' used in prior definitions of Δ by an $\mathcal{N}_e$-strategy $\beta \neq \alpha$.

(v) $Y(b) = 0$ where b is a code for the Σ_1^U-question "Is there a future stage at which the outcome of an $\mathcal{N}_\Phi$- or $\mathcal{P}_\Psi$-strategy β with $\beta\hat{\ }\langle n\rangle \subseteq \alpha$ (for some n) is $> n$ with a U-correct computation $\Phi(A\oplus U; n)$, or $\Psi(U)\restriction(c_n+1)$, respectively, or at which the outcome of an $\mathcal{N}_i$-strategy β with $\beta\hat{\ }\langle 0\rangle \subseteq \alpha$ is 1 with a U-correct computation $\Phi(A \oplus U; i)$?"

(vi) $Y(b') = 1$ for all such codes b' used in prior definitions of Δ by an $\mathcal{N}_e$-strategy $\beta \neq \alpha$.

(vii) $Y(d) = 0$ where d is a code for the Σ_1^U-question "Is there a future stage at which α is eligible to act and has a different guess about $(A \oplus U)'(e)$ (with a U-correct use if then $(A \oplus U)'(e) = 1$)?"; and

(viii) $Y(d') = 1$ for all such codes d' used before by α with a different guess about $(A \oplus U)'(e)$.

(Note that by Kleene's Fixed-Point Theorem, we may fix such indices b, c, and d during the construction. Note that the single computation $\Delta(S \oplus U'; e)$ is being defined by many different $\mathcal{N}_e$-strategies; the codes used in the oracle above ensure that there will be no incompatible definitions as verified in detail below; i.e., there may be several definitions using the same oracle but then they will also give the same value for $\Delta(S \oplus U'; e)$.)

At the end of each substage t, initialize all strategies $\beta >_L \alpha\hat{\ }\langle n\rangle$ where n is the current outcome of the strategy α which acted at substage t; furthermore, initialize all $\beta \geq \alpha\hat{\ }\langle n\rangle$ if α is a $\mathcal{P}_\Psi$- or $\mathcal{N}_\Phi$-strategy and n has entered V since the last substage s' at which $\alpha \subseteq f_{s'}$.

At the end of stage s, $\mathcal{R}$ redefines $\Theta(U \oplus V; x) = A(x)$ for all $x \leq s$ for which $\Theta(U \oplus V; x)$ is currently undefined. The use $\vartheta(x)$ is defined to be

(i) 0, if currently $x \in A$; or if x is currently not the coding location of any $\mathcal{P}$-strategy and not the use of a computation $\Gamma(A \oplus U; n, s') = 1$ for some s';

(ii) n, if $x \notin A$, x is currently the coding location c_n of a $\mathcal{P}$-strategy, and currently $n \notin V$; and

(iii) $w_s(n)$ if $x \notin A$, x is currently the use $\gamma(n, s')$ of a computation $\Gamma(A \oplus U; n, s') = 1$ for some s', and currently $n \in W(V)$.

Otherwise, x is the current coding location of a $\mathcal{P}$-strategy $\alpha <_L f_s$ for which $x \notin A$ but $n \in V$, or the use of a computation $\Gamma(A \oplus U; n, s') = 1$ for some s' for which $x \notin A$ but $n \notin W(V)$. In the former case set $\mathcal{A} = \{\alpha\}$, in the latter case let $\mathcal{A}$ be the set of all $\mathcal{P}_e$-strategies $\alpha <_L f_s$ for which $\alpha \subseteq f_{s''}$ for some $s'' \leq s'$ and α was not initialized since stage s''. (Note that $\alpha <_L f_s$ for all $\alpha \in \mathcal{A}$ by definition or since else α would have been initialized at stage s, or would have enumerated x into A at stage s.) For each $\alpha \in \mathcal{A}$, define $\beta(\alpha)$ to be the longest $\beta \subset \alpha, f_s$. The use $\vartheta(x)$ is then defined to be

(iv)

$$\max\Big\{\min\Big(\{\varphi_\beta(m) \mid \beta \text{ is an } \mathcal{N}_\Phi\text{-strategy with } \beta(\alpha) \subset \beta\hat{\ }\langle m\rangle \subseteq \alpha\} \cup \\ \{\psi_\beta(c_{\beta,m}) \mid \beta \text{ is a } \mathcal{P}_\Psi\text{-strategy with } \beta(\alpha) \subset \beta\hat{\ }\langle m\rangle \subseteq \alpha\} \cup \\ \{\varphi_e(e) \mid \beta \text{ is an } \mathcal{N}_e\text{-strategy with } \beta(\alpha) \subset \beta\hat{\ }\langle 0\rangle \subseteq \alpha\} \cup \\ \{w(e) \mid \beta \text{ is a } \mathcal{P}_e\text{-strategy with } \beta(\alpha) \subset \beta\hat{\ }\langle 0\rangle \subseteq \alpha\}\Big) \mid \alpha \in \mathcal{A}\Big\}.$$

(Note here that at least the use for $\beta(\alpha)$ in the above expression must be finite for each $\alpha \in \mathcal{A}$ since $\beta(\alpha)\hat{\ }\langle m\rangle <_L f_s$ for the corresponding m (or $m = 0$ for $\mathcal{P}_e$- or $\mathcal{N}_e$-strategies β). If $\mathcal{A}$ is empty then the use $\vartheta(x)$ will simply be set to 0 since no strategy can possibly enumerate x.)

Verification: Denote by $f \in [T] \cup T$ the true path of the construction, i.e., the lim inf of f_s as s tends to infinity. Recall that f may be of finite length, i.e., be a node on the tree with $\mathbf{0}'''$-outcome. (We then only need to verify the satisfaction of the requirements handled by strategies $\subseteq f$.) We proceed in two lemmas.

Lemma 1. *Requirement $\mathcal{R}$ is satisfied.*

Proof. Fix argument x of $\Theta(U \oplus V)$.

We first show that $\Theta(U \oplus V; x)$ is defined. Since $\mathcal{R}$ redefines $\Theta(U \oplus V; x)$ at all but finitely many stages at which it is undefined, we merely need to show that the use $\vartheta(x)$ is bounded. This is clear from the construction unless (i) x is the permanent coding location c_n of some $\mathcal{P}$-strategy α (which is thus not initialized after picking x), $n \in V$, and $x \notin A$; or (ii) x is the permanent use $\gamma(n, s')$ for some $n \notin W(V)$ and some s' for which $\Gamma(A \oplus U; n, s') = 1$. In case (i), fix the stage s_0 at which n enters V. Since $\alpha <_L f$, fix the longest node $\beta \subset \alpha, f$ as well as β's outcome m along α. Fix the (least) stage s_1 such that $f_s >_L \beta\hat{\ }\langle m\rangle$ for all $s \geq s_1$. But then $\varphi_\beta(m)$, $\psi_\beta(c_{\beta,m})$, $\varphi_{e_\beta}(e_\beta)$, or $w(e_\beta)$, respectively, is one of the uses in (iii) of the definition of $\Theta(U \oplus V; x)$ after stage s_1; so this use is bounded as desired. In case (ii), fix the least stage $s_0 > s'$ at which n leaves $W(V)$. Note that the set $\mathcal{A}$ in the definition of $\vartheta(x)$ must stabilize at some stage since there are only finitely many α's which are eligible to act before stage s'. Once $\mathcal{A}$ has stabilized by some stage $s_1 \geq s_0$, say, we see that all $\alpha \in \mathcal{A}$ are $<_L f_s$ for all $s \geq s_1$. Now we can argue as in case (i), noting that if $\mathcal{A} = \emptyset$ then $\vartheta(x)$ is set to 0.

Finally, we need to show that $\Theta(U \oplus V)$ correctly computes A. Fix an argument x and assume that $\Theta(U \oplus V; x) \neq A(x)$. Then x must be enumerated as (i) the coding location c_n by some $\mathcal{P}$-strategy α at a stage s_2 since at a stage $s_0 \leq s_2$, n entered V; or (ii) the use $\gamma(n, s')$ for some n and s' at a stage s_2 since at a stage $s_0 \leq s_2$, n has left $W(V)$. Since the coding location c_n or use $\gamma(n, s')$, respectively, is chosen at a stage $< x$, the use $\vartheta(x)$ is always chosen to be n or $w(n)$, respectively, before stage s_0, so at the end of stage s_0, $\Theta(U \oplus V; x)$ is redefined. Since by our assumption, $\Theta(U \oplus V; x) = 0$, this computation must become defined permanently before stage s_2, say at a stage $s_1 \in [s_0, s_2)$. Since x is not enumerated into A until stage s_2, and since α is not initialized between stages s_0 and s_2, $\alpha <_L f_{s_1}$ but $\alpha \subset f_{s_2}$. But all the computations $\Phi(A \oplus U; m)$, $\Psi(U)\restriction(c_{\beta,m})$, and $\Phi_{e_\beta}(e_\beta)$ and all the enumerations $W(V)(e_\beta)$ for the β's used in the definition of $\vartheta(x)$ at stage s_1 become undefined between stages s_1 and s_2. Thus $(U \oplus V)\restriction(\vartheta(x) + 1)$ changes between stages s_1 and s_2, contradicting the choice of s_1.

Remark. Note again that requirement $\mathcal{R}$ is ensured by the following feature of the construction: Let

$$S = \{\langle \alpha, s\rangle \mid \forall s' \geq s(\alpha <_L f_{s'})\}, \text{ and}$$
$$T = \{\langle \alpha, s\rangle \mid \forall s' \geq s(\alpha \leq f_{s'})\}.$$

Then S and T are both computably enumerable in $U \oplus V$ by our collapsing of the outcomes ∞_n and w_n into one outcome n, and our using outcomes 0 and 1 for the $\mathcal{N}_e$- and $\mathcal{P}_e$-strategies. Now $\mathcal{R}$ can correctly compute A since it can enumerate S and so predict whether a $\mathcal{P}$-strategy α that is ready to enumerate its coding location into A will ever be on the current true path again. We remark that the above also shows that the true path is d-computably enumerable in $U \oplus V$.

We now turn to the satisfaction of the $\mathcal{P}$- and $\mathcal{N}$-requirements by strategies along the true path of the construction.

Lemma 2. *Every strategy $\alpha \subseteq f$ ensures the satisfaction of its requirement. If $\alpha = f$ then α contradicts the theorem's hypothesis.*

Proof. Fix $s_0 \geq |\alpha|$ least such that $\alpha \leq f_s$ and α is not initialized at stage s for all $s \geq s_0$. We now distinguish cases for α.

Case 1: α is a $\mathcal{P}_\Psi$-strategy: Then α will clearly satisfy its requirement as in the basic module of the strategy described above.

Case 2: α is an $\mathcal{N}_\Phi$-strategy: Then α will try to satisfy its requirement as in the basic module of the strategy described above. The only possible complication is that α's A-restraint for preserving a computation $\Phi(A \oplus U; n)$ may be injured. By initialization, only a $\mathcal{P}$-strategy β comparable with $\alpha\hat{}\langle n\rangle$ can possibly injure that restraint. First consider a $\mathcal{P}$-strategy such that $\beta\hat{}\langle m\rangle \subseteq \alpha$ for some m. When $\mathcal{P}_\Psi$-strategy β enumerates its coding location $c_{\beta,m'}$ for some $m' \leq m$ then α is initialized contrary to assumption on s_0; and any coding location $c_{\beta,m'}$ for $m' > m$ is chosen after α's A-restraint is imposed, so is chosen greater than that A-restraint. When a $\mathcal{P}_e$-strategy enumerates a use $\gamma(e, s')$ then it takes outcome 0, so if $\alpha \supseteq \beta\hat{}\langle 1\rangle$ then α is initialized at that stage, and if $\alpha \supseteq \beta\hat{}\langle 0\rangle$ then α's restraint is less than $\gamma(e, s')$ since all uses $\leq \alpha$'s restraint would already have been enumerated. Now consider the case $\beta \supseteq \alpha\hat{}\langle n\rangle$. Then β is eligible to act at a time when α imposes A-restraint for preserving a computation $\Phi(A \oplus U; n)$ only if n has just entered V but $U\restriction(\varphi(n)+1)$ has not changed yet. But then β would have been initialized after n enters V and so cannot injure α.

Case 3: α is an $\mathcal{N}_e$-strategy: We first show that α can always define $\Delta(S \oplus U'; e)$ according to its strategy unless α tries to define $\Delta(S \oplus U'; e) = 1$ while it sees a U-incorrect computation $\Phi_e(A \oplus U; e)$, or α is initialized later (in which case α's attempt to redefine is irrelevant anyhow). Suppose this fails due to a prior definition of $\Delta(S \oplus U'; e)$ to a different value by a strategy (necessarily an $\mathcal{N}_e$-strategy) β at a stage s'. If $\beta > \alpha$ then $\beta >_L \alpha$ and the current true path has thus moved to the left of β since stage s', so β's code c is one of the codes c' used by α. If $\beta = \alpha$ then α has changed its guess about $(A \oplus U)'(e)$, and if α now has a U-correct computation $\Phi(A \oplus U; e)$ in case it wants to set $\Delta(S \oplus U'; e) = 1$), then β's code d is now one of the codes d' used by α. Finally, if $\beta < \alpha$ then $\beta <_L \alpha$. Fix the node ξ at which β and α split. ξ cannot be a $\mathcal{P}_i$-strategy since then β and α would disagree about S. Otherwise, if α is not initialized later, then ξ permanently switches outcome to the right of β's guess about ξ's outcome, and so β's code d will be one of α's codes d', thus allowing α to redefine $\Delta(S \oplus U'; e)$.

Now let s_1 be the least stage $\geq s_0$ at which α finds a computation $\Phi_e(A \oplus U; e)$ (if such a stage exists), and let $s_1 = s_0$ otherwise. Then at stage s_1, β will define $\Delta(S \oplus U'; e) = (A \oplus U)'(e)$ with correct $S \oplus U'$-oracle (and thus define $\Delta(S \oplus U'; e)$ permanently). Finally, no strategy can injure α's A-restraint after stage s_1 since no $\beta <_L \alpha$ is eligible to act after stage s_0, no β with $\beta\hat{}\langle m\rangle \subseteq \alpha$ for some m enumerates a number $\leq s_1$ after stage s_1 (since it will either have enumerated all possible numbers $\leq s_1$ by stage s_1 already, or it will else α will be initialized), and no β with $\alpha < \beta$ enumerates a number $\leq s_1$ after stage s_1 (since it be initialized before first acting at a stage $\geq s_1$). Thus $(A \oplus U)'(e) = \Delta(S \oplus U'; e)$ as desired.

Case 4: α is a $\mathcal{P}_e$-strategy: Then α can reset $\Gamma(A; e, s) = 0$ (if it so desires) for almost all s.

6. $\mathbf{0}'''$- or Π_3-Constructions

6.1. General remarks. In $\mathbf{0}'''$- or Π_3-constructions, we usually have strategies α measuring some Π_2-statement (typically whether there are infinitely many expansionary stages for some computations), which in turn control an infinite number of substrategies β working on subrequirements of α's requirement. If some substrategy β has a Π_2-outcome, this typically indicates a Σ_3-outcome for the overall requirement and that we do not need any more substrategies for α below that outcome of β. In the other case, each substrategy β has a Σ_2-outcome, and α and all its substrategies together ensure the satisfaction of α's requirement in a Π_3-fashion. (Often the substrategies β simultaneously work for another requirement, and their failure to satisfy this other requirement gives rise to a Σ_3-outcome for α.)

More precisely, a typical Π_3-requirement has the form

$$P \wedge R \to S$$

where P is a Π_2- and $R = \exists n R_0(n)$ and $S = \exists n S_0(n)$ are Σ_3-conditions, possibly relative to some computably enumerable oracles which we are building. Then the main strategy α will measure P while the substrategies β measure the Π_2-condition $R_0(n)$ (for some n) and ensure the Π_2-condition $S_0(n)$ if necessary, very much the way a Π_2-strategy would act independently.

Often, a Π_2-outcome of a substrategy β will injure the action of substrategies β' working for a main strategy α' of lower priority than α. This will mean that we have to introduce a new main strategy α'' below the Π_2-outcome of β which works for the same requirement as α' (and which then, of course, has its own new substrategies β''). This corresponds to "finite injury" along the true path of the construction, i.e., if we look along the true path of the construction, then the behavior of the main strategies is analogous to the behavior of strategies in a finite injury argument (without using a tree of strategies), where nodes along the true path in the $\mathbf{0}'''$-construction correspond to stages in the finite injury argument without tree of strategies. (It is possible to avoid this finite injury along the true path and have a single strategy work on the entire Π_3-requirement by itself; however, this then makes the behavior of the true path and the initialization much more complicated.)

6.2. The Lachlan Nonbounding Theorem. This theorem is probably the easiest $\mathbf{0}'''$-priority argument. It shows that the joins of minimal pairs are not downward dense (whereas halves of minimal pairs, i.e., the *cappable* degrees, are).

Theorem 20. (Lachlan Nonbounding Theorem (1979)) *There is a noncomputable computable enumerable degree* $\mathbf{a}$ *such that there is no minimal pair of computably enumerable degrees* $\mathbf{u}, \mathbf{v} \leq \mathbf{a}$.

6.3. The Slaman Density Theorem. This theorem combines the techniques of the Sacks Density Theorem with the Fejer-Lachlan technique for building branching degrees in a very intricate way, using so-called configurations. It establishes that the diamond lattice may be embedded densely into the computably enumerable degrees.

Theorem 21. (Slaman Density Theorem (1989?)) *Given any computable enuerable degrees* $\mathbf{u} < \mathbf{v}$*, there are computably enumerable degrees* $\mathbf{a}, \mathbf{b}, \mathbf{c}$ *between* $\mathbf{u}$ *and* $\mathbf{v}$ *such that* $\mathbf{a}$ *and* $\mathbf{b}$ *are incomparable and meet to* $\mathbf{c}$*. (Thus the diamond lattice (formed by* $\mathbf{a} \cup \mathbf{b}$*,* $\mathbf{a}$*,* $\mathbf{b}$*, and* $\mathbf{c}$*) can be embedded densely into the computably enumerable degrees.*

6.4. The Lachlan Nonsplitting Theorem. This theorem is the $\mathbf{0}'''$-priority argument. It shows that the Sacks Density Theorem and the Sacks Splitting Theorem cannot be combined.

Theorem 22. (Lachlan (1979)) *There are computable enumerable degrees* $\mathbf{b} < \mathbf{a}$ *such that* $\mathbf{a}$ *does not split over* $\mathbf{b}$*, i.e., there are no incomparable computably enumerable degrees* $\mathbf{u}, \mathbf{v} \geq \mathbf{b}$ *with* $\mathbf{u} \cup \mathbf{v} = \mathbf{a}$.