ISRAEL JOURNAL OF MATHEMATICS **137** (2003), 285–320

THERE EXISTS A MAXIMAL 3-C.E. ENUMERATION DEGREE*

BY

S. BARRY COOPER

Department of Pure Mathematics, School of Mathematics
University of Leeds, Leeds LS2 9JT, U.K.
e-mail: pmt6sbc@leeds.ac.uk

AND

ANGSHENG LI

Institute of Software, Academia Sinica
Beijing, 100080, China
e-mail: angsheng@ios.ac.cn

AND

ANDREA SORBI

Dipartimento di Scienze Matematiche ed Informatiche "Roberto Magari"
Via del Capitano 15, 53100 Siena, Italy
e-mail: sorbi@unisi.it

AND

YUE YANG

Department of Mathematics, Faculty of Science
National University of Singapore, Lower Kent Ridge Road, Singapore 119260
e-mail: matyang@math.mus.edu.sg

* The first two authors were partially supported by EPSRC Research Grant "Turing Definability" No. GR/M 91419 (UK), and the second author by NSF grant No. 69973048 and by NSF major grant No. 19931020 (P. R. China), and by an INDAM visiting professorship at the University of Siena. The fourth author was partially supported as a visiting scholar by the University of Siena. The first three authors were funded by the INTAS-RFBR joint project *Computability and Models*, no. 972–139. The fourth author would like to thank Marat Arslanov for useful discussions.
Received May 24, 2001

ABSTRACT

We construct an incomplete 3-c.e. enumeration degree which is maximal among the n-c.e. enumeration degrees for every n with $3 \leq n \leq \omega$. Consequently the n-c.e. enumeration degrees are not dense for any such n. We show also that no low n-c.e. e-degree can be maximal among the n-c.e. e-degrees, for $2 \leq n \leq \omega$.

1. Introduction

Computability theory has until recently been dominated by the structure of the Turing degrees, especially of the computably enumerable (or c.e.) Turing degrees. As the structure of the c.e. Turing degrees becomes increasingly well understood, leaving a small number of important and intractable problems apparently out of the reach of current techniques, there is more and more interest in related structures — such as the n-c.e. Turing degrees, the enumeration degrees of the Σ_2^0 sets and the n-c.e. enumeration degrees, each of which forms a natural extensions of the c.e. Turing degrees. Besides their theoretical interest, compared with the Turing or c.e. Turing degrees these extensions appear to have many novel features, including that they are based on reducibilities provided by functionals that need not be total. An algorithm Φ relative to auxiliary information (or oracle) K yields a function Φ^K computable from K, which may or may not be defined on all inputs. Restricting one's attention to the special case when Φ^K is a characteristic function leads directly to the sets Turing reducible to K, and to the familiar local structure of the Turing degrees. The basis for the exclusion of nontotal Φ^A is, of course, nonconstructive (by Rogers [19], the set of indices of total Φ^A is Π_2^A-complete). And in any case, there are theoretical imperatives leading not merely to a more **comprehensive** structure (the enumeration degrees), but to a mathematically informative context for the Turing degrees themselves. This is the perspective from which we approach, below, the question of local density/nondensity of degree structures. We begin with a brief survey of density results for various degree structures.

1.1. The Turing degrees. A prototypical result of Sacks in the early 1960's settled the density problem for the c.e. Turing degrees.

Theorem 1.1 (Sacks Density Theorem [22]): *For every pair of c.e. Turing degrees $\mathbf{a} < \mathbf{b}$, there is a c.e. Turing degree $\mathbf{c}$ such that $\mathbf{a} < \mathbf{c} < \mathbf{b}$.*

Concerning the global structure of the Turing degrees, Spector [23] showed that there is a minimal degree (below $\mathbf{0}''$), a result later improved by Sacks.

THEOREM 1.2 (Sacks [21]): *There is a minimal degree* $\mathbf{a} < \mathbf{0}'$.

1.2. THE ENUMERATION DEGREES. Let us first recall some basic facts about the enumeration degrees. Our formalization of enumeration reducibility closely follows [12]; see also [20]. For more information, see Cooper's survey paper [6].

Definition 1.3: An **enumeration operator** (or **e-operator**) is a computably enumerable set Ψ of pairs $\langle x, F\rangle$, where F is (the code of) a finite subset of ω. Each pair $\langle x, F\rangle \in \Psi$ is called a **Ψ-axiom** and F is called a **positive neighborhood condition.** For a set B, if $F \subseteq B$ then we say that the Ψ-axiom $\langle x, F\rangle$ **applies to** B. For any e-operator Ψ and any set B, Ψ^B is defined to be the set:

$$\Psi^B = \{x : (\exists F)(\langle x, F\rangle \in \Psi \wedge F \subseteq B)\}.$$

We say that a set A is *e-reducible* to a set B (in symbols: $A \leq_e B$) if there is an e-operator Ψ such that $A = \Psi^B$.

The degree structure induced by e-reducibility is the structure of the **enumeration degrees** (or **e-degrees**), which is a natural extension of the structure of the Turing degrees, a fact first noticed by Myhill [17]. Indeed, it is easily seen that the mapping $\deg_T(A) \mapsto \deg_e(c_A)$ (where c_A denotes the graph of the characteristic function of A) is an order-theoretic embedding of the Turing degrees into the e-degrees. See for instance [20, Corollary 9.XXIV].

It is easy to see that under this embedding the c.e. Turing degrees correspond to the Π^0_1 e-degrees. Thus the Sacks Density Theorem for the c.e. Turing degrees immediately gives us density of the Π^0_1 e-degrees.

Density in the e-degrees is by now a fairly well understood phenomenon. It is known for instance that the local structure of the e-degrees, which coincides with the Σ^0_2 e-degrees, is dense.

THEOREM 1.4 (Cooper [4]; see also Lachlan and Shore [15]): *The structure of the Σ^0_2 e-degrees is dense.*

Unlike the case of the Turing degrees, there is no minimal e-degree, as proved by Gutteridge, [13]; see also Cooper [4]. The proof combines two results: First, any candidate to be a minimal e-degree must be Δ^0_2; then it makes use of the following lemma.

LEMMA 1.5 ([5]): *For any Δ^0_2 set B with $\emptyset <_e B$, there is a Δ^0_2 set A such that $\emptyset <_e A <_e B$.*

However, the global structure of the e-degrees is not dense. This result was first shown by Cooper [5], and later improved by Calhoun and Slaman.

THEOREM 1.6 (Calhoun and Slaman [3]): *The structure of the Π^0_2 e-degrees is not dense. In fact there is an empty interval* $(\mathbf{a}, \mathbf{b})$ *of e-degrees with* $\mathbf{a}$ *and* $\mathbf{b}$ Π^0_2 *e-degrees and* $\mathbf{a} <_e \mathbf{b}$.

Recently, Arslanov, Kalimullin and Sorbi established the full density of the Δ^0_2 e-degrees.

THEOREM 1.7 (Arslanov, Kalimullin and Sorbi [2]): *The structure of the Δ^0_2 e-degrees is dense.*

1.3. The n-c.e. Turing degrees.

The n-c.e. sets were first studied by Putnam [18] and Ershov [9], [10], [11]. For more information, see e.g. Epstein, Haas and Kramer [8].

We recall the basic definitions.

Definition 1.8: A set A is said to be n-**c.e.** if there is a computable function f such that for all x, $A(x) = \lim_s f(x,s)$, $f(x,0) = 0$ and

$$(*) \qquad |\{s : f(x,s) \neq f(x,s+1)\}| \leq n,$$

i.e. $f(x,s)$ can change at most n times before reaching its limit.

A is said to be ω-c.e. if $(*)$ is replaced by

$$(**) \qquad |\{s : f(x,s) \neq f(x,s+1)\}| \leq x.$$

2-c.e. sets are often called **d.c.e. sets**. Clearly, the 1-c.e. sets are just the c.e. sets.

By an unpublished result of Lachlan, every noncomputable n-c.e. Turing degree bounds a noncomputable c.e. Turing degree. Consequently the structure of the n-c.e. Turing degrees is downward dense by the Sacks Density Theorem. On the other hand, the ω-c.e. Turing degrees are not dense: This can be seen by observing that the minimal degree constructed in the proof of Theorem 1.2 is in fact ω-c.e.

The upward density of the n-c.e. Turing degrees remained elusive for a long time. Eventually, joint work of Lachlan and Soare towards a negative result led to the full solution which appeared in Cooper, Harrington, Lachlan, Lempp and Soare [7].

THEOREM 1.9 (Cooper et al. [7]): *There is a maximal incomplete d.c.e. Turing degree.*

In fact it is shown in [7] that there exists an incomplete d.c.e. Turing degree which is maximal among the n-c.e. (with $n \geq 2$) and ω-c.e. Turing degrees.

Consequently the structures of the n-c.e. (for $n \geq 2$) and of the ω-c.e. Turing degrees are not dense.

1.4. THE n-C.E. ENUMERATION DEGREES. Not until the late 1990's was there any serious investigation of the structure of the n-c.e. e-degrees. One example is the recent result by Kalimullin [14], refuting Downey's conjecture in the e-degrees. This result states that it is not the case that the structures of the n-c.e. e-degrees are all elementarily equivalent, for $n \geq 2$.

We now turn to density problems in the n-c.e. e-degrees. Since every d.c.e. e-degree contains a Π_1-set, the structure of the d.c.e. e-degrees is dense. Also by examining the proof of Lemma 1.5, one can show that the structure of the ω-c.e. e-degrees is downward dense. Recently, Arslanov, Kalimullin and Sorbi, [3], proved that the structure of the n-c.e. e-degrees (for $n \geq 3$) is downward dense: In fact every nontrivial n-c.e. e-degree, for $n \geq 3$, bounds a nontrivial 3-c.e. e-degree.

This leads us to the only remaining density problem in the n-c.e. e-degrees, raised by Arslanov at the Boulder meeting on open problems in computability theory (see [1, Open Problem 3.6]): Are the n-c.e. e-degrees dense for each $n \geq 3$? Are the ω-c.e. e-degrees dense?

In this paper, we establish the following results which give negative answers to the questions raised by Arslanov at the Boulder meeting.

THEOREM 1.10: *There is a maximal incomplete* 3-*c.e. e-degree, i.e., a* 3-*c.e. e-degree* $\mathbf{c} < \mathbf{0}'_e$ *such that there is no* 3-*c.e. e-degree* $\mathbf{e}$ *with* $\mathbf{c} < \mathbf{e} < \mathbf{0}'_e$. *Thus the partial order of the* 3-*c.e. e-degrees is not dense.*

(We recall that $\mathbf{0}'_e$ is the greatest element of the local structure of the e-degrees.) As in the Turing degrees, the result can be generalized to the structures of the n-c.e. or the ω-c.e. e-degrees.

THEOREM 1.11: *There is an incomplete* 3-*c.e. e-degree which is maximal in the* n-*c.e. e-degrees (for all* $n \geq 3$*) as well as in the* ω-*c.e. e-degrees. Thus the partial orders of the* n-*c.e. e-degrees (for* $n \geq 3$*) and the partial order of the* ω-*c.e. e-degrees are not dense.*

However (Corollary 6.2) we also prove that no low n-c.e. e-degree can be maximal among the n-c.e. e-degree for $2 \leq n \leq \omega$.

1.5. THE PLAN OF THE PAPER. Our plan for showing the existence of a maximal 3-c.e. e-degree is to suitably adapt the proof of Theorem 1.9 from [7] to the context of the e-degrees. However, the differences between Turing reducibility and enumeration reducibility give rise to many subtle points. For those who are familiar with [7], we briefly mention some of these points here. The discussion will be informal, because many notations will be formally introduced only in the following sections.

We want to construct a 3-c.e. set C with maximal 3-c.e. and incomplete e-degree. An axiom in an e-operator has no negative neighborhood conditions, thus, a computation can only be killed by destroying its positive neighborhood condition, that is, by extracting numbers from the oracle. This leads us to set up C_0 to be ω at stage 0. Extracting a number and putting it back, as dictated by the action, results in a 3-c.e. set C instead of a d.c.e. set as in the Turing case.

Lack of negative neighborhood conditions also makes us treat Γ^{CU} and Δ^C differently. When we build two e-operators Γ and Δ, trying to achieve $\Gamma^{CU} = \overline{K}$ and $\Delta^C = U$, where U is a given 3-c.e. set, in order to satisfy an $\mathcal{S}$-requirement, we hope to have the following dichotomy: If U changes then we make progress on Γ; otherwise, we make progress on Δ. This is what happens in the proof of Theorem 1.9 in [7]. Now in the e-degrees, if some element re-enters U, then we cannot make use of this change to progress Γ, since any positive neighborhood condition still applies to $C \oplus U$, and we have to correct (in fact, update) Δ by adding more axioms. As regards Γ^{CU}-uses, when some element w leaves U, there might be some R-controller strategy that wants to use this change to lift the $\Gamma^{CU}(w')$-use for some $w' > w$. Thus we make the Γ-use function to be increasing.

Finally, we want to point out that an e-operator Δ (for which we want, say, $\Delta^C = U$) only requires a Δ-axiom $\langle x, F\rangle$ to apply to C for $x \in U$. If $x \notin U$, then Δ may not contain any axioms of the form $\langle x, F\rangle$ for $F \subseteq C$ (this corresponds to the undefined case in the Turing degrees). In this sense, Δ^C is always total. This makes us modify the selection of the killing point for $\Delta^C = U$. The problem occurs when the 3-c.e. set U is finite. In that case, Δ could either be a finite set (of axioms), so that we might never see any candidate for being a killing point; or the candidate v for being the killing point keeps moving to infinity. However, this works to our advantage. All we need to do is to find a threshold u and kill the computations $\Delta^C(v)$ for $v \geq u$, (regardless to whether we cope with the same v or not).

Let us list a few conventions. Given an enumeration operator Θ, we use the corresponding lower case letter $\theta^X(x)$ (or $\theta(x)$ if the oracle is clear from the context) to denote the use function for $\Theta^X(x) = 1$. If Θ is not built by us (in dealing with Θ^C where C is built by us and Θ^C is supposed to compute a Π^0_1 set) we agree that $\theta(x)$ is the least number such that no computation $\Theta^C(y) = 1$, with $y \le x$, is destroyed by extracting any number $z \ge \theta(x)$. (Thus the use is increasing with respect to the argument x.) Of course the function θ need not be total.

In the case of e-operators built by us, such as the e-operators Γ's and Δ's, we will agree that the use functions will coincide with certain (partial) functions γ and δ, respectively, which will be provided by the construction.

Given any set X and $x \in \omega$, then we let

$$X \upharpoonright x = \{y : y \in X : y < x\}.$$

If the e-operator Γ applies to the join of two sets X and Y, we will write Γ^{XY} instead of $\Gamma^{X\oplus Y}$ and we also assume that the use is computed in the two sets separately, i.e., $\Gamma^{XY\upharpoonright(\gamma(x)+1)}$ will mean $\Gamma^{X\upharpoonright(\gamma(x)+1)\oplus Y\upharpoonright(\gamma(x)+1)}$.

During the course of a construction, whenever we define a parameter as **fresh** or **big**, we mean that it is defined as the least natural number which is greater than any number mentioned so far in the construction.

Finally, we assume that the tree of strategies grows upwards.

2. The requirements and the basic strategies

We want to construct a 3-c.e. set $C <_e \overline{K}$ such that there is no 3-c.e. set U with $C <_e U <_e \overline{K}$. We recall that the e-degree of $\overline{K}$ (i.e. the complement of the halting set K) is the greatest element of the local structure. Fix an effective enumeration of all 3-c.e. sets $\{U_e\}_{e\in\omega}$. For each U_e, we build an enumeration operator Γ_e satisfying the following requirement:

$\mathcal{S}_e$: $\overline{K} = \Gamma_e^{CU_e}$ or $(\exists \Delta_e)$ $(U_e = \Delta_e^C)$.

Fix an effective enumeration of all e-operators $\{\Theta_e\}_{e\in\omega}$. To ensure that C is incomplete, we build an auxiliary Π^0_1 set A satisfying the following requirements:

$\mathcal{R}_e$: $A \neq \Theta_e^C$.

In the rest of this section, we will describe the basic modules for $\mathcal{R}$-destroyer-strategies, $\mathcal{R}$-controller strategies and $\mathcal{S}$ strategies. In the next section, we run a test case for two $\mathcal{S}$-strategies to illustrate the general ideas. For more intuition and for the evolution of the ideas about $\mathcal{R}$-destroyer and $\mathcal{R}$-controller strategies, see Cooper et al. [7].

2.1. THE BASIC $\mathcal{S}$-STRATEGY. The basic task for $\mathcal{S}$ is to build an e-operator Γ such that Γ^{CU} computes $\overline{K}$. The job for building Δ is left for some lower priority $\mathcal{R}$-destroyer strategies. We begin with $C_0 = \omega$. As time goes by, $\mathcal{S}$ defines $\Gamma^{CU}(w) = 1$ with use $\gamma(w) \in C$ for more and more w's. When w enters K, we extract $\gamma(w)$ from C to preserve the correctness of Γ.

We now make some remarks on uses, which will make clear what we mean by use functions for Γ and Δ. As in [7], when we define $\Gamma^{CU}(w)$ with use $\gamma(w)$, we not only enumerate the axiom

$$\langle w, C \restriction (\gamma(w)+1) \oplus U \restriction (\gamma(w)+1)\rangle$$

into Γ, but we also reserve an interval, called a **use block**,

$$B(w) = [\gamma(w) - n, \gamma(w)]$$

(for some n), solely for destroying and restoring $\Gamma^{CU}(w)$. The choice of $|B|$ (the size of the block B) will be discussed later. We assume that $|B|$ is less than the least element of the block B. We also assume that whenever some element needs to be extracted from a use block, it will be the least unused element of it. We make a similar convention for Δ^C-uses.

Initially, we choose the use $\gamma(w)$ for $\Gamma^{CU}(w)$ to be big. When the oracle $C \oplus U$ changes below $\gamma(w)$ and no previous definition of $\Gamma^{CU}(w)$ applies, we redefine $\Gamma^{CU}(w)$ as follows: First note that for this to happen, some element must leave C or U. If no element in the use block $B(w')$ with $w' \leq w$ has left from C, then we will redefine $\Gamma^{CU}(w)$ with the same use $\gamma(w)$; otherwise, we will redefine $\Gamma^{CU}(w)$ with big use. This is to prevent other Δ- or Γ'-uses from interfering with the definition of $\Gamma(w)$. Summarizing: $\gamma(w)$ only moves if for some $w' \leq w$, $B(w')$ changes; in all other cases, $\gamma(w)$ remains the same.

The $\mathcal{S}$-strategy may be injured if some lower priority $\mathcal{R}$-destroyer strategy $\mathcal{R}$ builds an e-operator Δ such that $U = \Delta^C$. In the process of building Δ, there would be a point $w \in \overline{K}$ such that all definitions of $\Gamma^{CU}(w')$ with $w' \geq w$ are destroyed by $\mathcal{R}$, i.e., no $\langle w', F\rangle \in \Gamma$ will apply to $C \oplus U$. In this case, the requirement $\mathcal{S}$ will be satisfied by the $\mathcal{R}$-destroyer strategy, which we discuss next.

2.2. THE BASIC $\mathcal{R}$-DESTROYER STRATEGY. The task for an $\mathcal{R}$-destroyer strategy is to either successfully diagonalize against its Θ or build $\Delta^C = U$ for some higher priority $\mathcal{S}$-strategy. We try to accomplish the first task via a simple Friedberg–Muchnik strategy. We start with $A_0 = \omega$:

(1) Pick a fresh witness $x \in A$ and keep it in A.
(2) Wait for a computation $\Theta^C(x) = 1$ which is cleared of all higher priority γ- and δ-uses. (We say that a computation $\Theta^C(x)$ is **cleared of** the use $\gamma(w)$ if $\theta(x)$ is less than the least element of the use block for $\Gamma^{CU}(w)$. Similarly for δ-uses.)

(3) If we find one, then extract x from A and restrain $C \upharpoonright (\theta(x)+1)$, i.e., do not allow elements in this set to leave C.

If no such computation is found, then we perform the so-called "capricious destruction" and turn to the task of building Δ for some higher priority requirement $\mathcal{S}$.

We define $\Delta^C = U$ as follows. As time goes by, we pick the least number v in U such that $\Delta^C(v)$ is undefined and define $\Delta^C(v) = 1$ with big use $\delta(v)$, that is, enumerate $\langle v, C \upharpoonright (\delta(v)+1)\rangle$ into Δ and specify a use block similar to what we did for Γ. In normal circumstances, when v leaves U, we correct $\Delta^C(v)$ by extracting an element from the $\delta(v)$-use block. If later v comes back again into U, then we redefine $\Delta^C(v) = 1$ with big use. Note that in this case we could safely redefine $\Delta^C(v) = 1$ with $\delta(v) = 0$ by adding $\langle v, \emptyset\rangle \in \Delta$. We prefer however to redefine $\Delta^C(v) = 1$ with big use, to avoid having tediously to distinguish during the construction between elements entering U for the last time and elements entering U for the first time. Note also that Γ has no such feature as $\overline{K}$ in Π^0_1.

However, in the case when this $\mathcal{R}$-destroyer strategy is taken charge of by some $\mathcal{R}$-controller strategy, we will follow the instructions imposed by the controller, which will be discussed later.

In Section 4 we will use a binary tree T to organize strategies. If a strategy is assigned to a node β of T, then it is convenient to identify the strategy with the node β corresponding to it. The informal discussions that follow frequently refer to this identification.

We now discuss an $\mathcal{R}$-destroyer strategy β in more detail. In general, β will have to deal with a finite (nonzero) number of e-operators Γ's built by higher priority $\mathcal{S}$-strategies (and not destroyed yet), and a finite number of Δ's built by higher priority $\mathcal{R}$-destroyer strategies (and not destroyed yet). β will destroy the lowest priority one of the Γ's, say $\overline{\Gamma}$, and also the Δ's of lower priority than $\overline{\Gamma}$, and it will build $\overline{\Delta}^C = U$.

An $\mathcal{R}$-destroyer strategy β has a fixed "killing point" w for killing $\overline{\Gamma}$. β has also a threshold value u for destroying $\Delta^C = U$. (Note that we could take u to be w, but we want to emphasize the asymmetry between w and u.) Whenever $\overline{K} \upharpoonright w$ or $U \upharpoonright u$ changes (that is, in the latter case, some element $v < u$ either enters or leaves U), we discard the e-operator $\overline{\Delta}$ which β is currently building and start a new copy instead. The strategy thus assumes that $\overline{K} \upharpoonright w$ and $U \upharpoonright u$ have already settled down, which is true after a finite amount of injury.

2.3. THE BASIC $\mathcal{R}$-CONTROLLER STRATEGY. If there is no Γ to destroy, then the $\mathcal{R}$-destroyer strategy becomes an $\mathcal{R}$-controller one. An $\mathcal{R}$-controller strategy

γ picks a witness x and waits for computations for its own Θ and all e-operators Θ_β relative to higher priority $\mathcal{R}$-destroyer strategies β with infinite outcome (including the ones whose Δ_β has been destroyed already), and for the use-blocks of the e-operators Γ_β and Δ_β to be sufficiently large with respect to $\theta(x)$ and all the $\theta_\beta(x)$'s. If the $\mathcal{R}$-controller fails to find computations as specified, then the corresponding requirement $\mathcal{R}$ is satisfied without taking any action, otherwise the $\mathcal{R}$-controller will **take charge of all the higher priority $\mathcal{R}$-destroyer strategies** β and ensure that either $\mathcal{R}$ itself or one of these higher priority strategies β satisfies its corresponding requirement by diagonalization.

3. A simple case: Working above two $\mathcal{S}$-strategies

We run through the case where two $\mathcal{S}$-strategies are present. It will give us an intuition for the general case. We also use this opportunity to introduce some notations and discuss some fine points which have been left out in the previous section.

Assume that α_0 and α_1 are $\mathcal{S}$-strategies working for the requirements $\mathcal{S}_0$ and $\mathcal{S}_1$, and building e-operators Γ_0 and Γ_1 respectively. Assume that $\alpha_0 \subset \alpha_1$, in other words $\mathcal{S}_0$ has higher priority than $\mathcal{S}_1$. We describe the $\mathcal{R}$-strategies above them one by one.

3.1. Description of an $\mathcal{R}_0$-destroyer strategy. The first node β_0 above α_1 will be an $\mathcal{R}_0$-destroyer strategy. The goal for β_0 is either to find a Θ_0-computation which is cleared of both γ_0- and γ_1-uses; or to destroy $\Gamma_1^{CU_1}$ and build $\Delta_1^C = U_1$.

The $\mathcal{R}_0$-destroyer strategy β_0 has three parameters: x_0 (called a **witness**) for diagonalization against Θ_0^C; w_0 (called a **killing point**) for destroying Γ_1; and a counter i_0 to record the number of previous Γ_1-killings performed by β_0 so far.

Initially, x_0 and w_0 are chosen fresh and i_0 is set to 0. We make some remarks on the choice of w_0. First, whenever w_0 enters K, we discard w_0 and take the least number greater than w_0 in (the current approximation to) $\overline{K}$ as the new killing point. As $\overline{K}$ is infinite, eventually we will find a killing point in $\overline{K}$. Secondly, when $\overline{K} \upharpoonright w_0$ changes, the strategy β_0 gets **reset**, that is, all previous $\mathcal{R}_0$-strategy actions are cancelled, but the killing point w_0 remains unchanged. From now on, we always assume that any killing point w for destroying some e-operator Γ is chosen from $\overline{K}$ and $\overline{K} \upharpoonright w$ never changes, which is true after finite injury. Having said this, we would like to point out that the fact that w_0 lies in $\overline{K}$ is not relevant. For example, we can destroy Γ as we destroy Δ, namely, by specifying a threshold u and killing all computations $\Gamma^{CU}(v) = 1$ for $v \geq u$.

However, choosing w_0 from $\overline{K}$ makes things closer to the Turing case, and offers us a desirable asymmetry between Γ and Δ.

The $\mathcal{R}_0$-destroyer strategy β_0 proceeds as follows.

1. Wait for $\Theta_0^C(x_0) = 1$ and
$$(\forall z \leq i_0)(z \in A \Rightarrow \Theta_0^C(z) = 1).$$
(The second condition is to help the future $\mathcal{R}$-controller strategy and slows down the $\mathcal{R}_0$-destroyer strategy.)
2. If $\min[\gamma_0(w_0)-, \gamma_1(w_0)$ – use blocks] is larger than $\theta_0(x_0)$ then extract x_0 from A, restrain $C \upharpoonright (\theta_0(x_0)+1)$, and stop.
3. Otherwise, define $B_{i_0}^1$ to be the current $\gamma_1(w_0)$-use block; extract $\gamma_1(w_0)$ from C; request that the $\mathcal{S}_1$-strategy choose the new $\gamma_1(w_0)$-use block to be very big; for any $i \leq i_0$ if $i \in U_1$ then define $\Delta_1^C(i) = U_1(i)$ with fresh use $\delta_1(i)$ and keep it correct from now on (unless stopped), and go back to 1. Notice that x_0 has not yet been extracted from A.

If we loop through 3. infinitely often, then in the process the use $\gamma_1(w_0)$ is lifted further and further to infinity.

The outcomes of the $\mathcal{R}_0$-destroyer strategy: β_0 has a finite outcome (denoted by 1), if it is eventually always in 1. or in 2. It has an infinite outcome (denoted by 0) if it goes from 3. to 1. infinitely often.

3.2. Description of an $\mathcal{R}_1$-destroyer strategy. Above the finite outcome of β_0, there will be an $\mathcal{R}_1$-destroyer strategy which proceeds exactly as the $\mathcal{R}_0$-destroyer strategy. Above the infinite outcome 0 of β_0, there will be an $\mathcal{R}_1$-destroyer strategy β_1. The plan for β_1 is either to find a Θ_1-computation cleared of both γ_0- and δ_1-uses; or to destroy both $\Gamma_0^{CU_0}$ and Δ_1^C and build $\Delta_0^C = U_0$.

The $\mathcal{R}_1$-destroyer strategy β_1 has the following parameters: A fresh witness x_1 for Θ_1; a killing point $w_1 > w_0$ from $\overline{K}$, a **threshold value** u_1 and a counter i_1 to record the number of previous Γ_0-killings performed by β_1. Similarly to the remark about the parameter w, if $U_1 \upharpoonright u_1$ changes, then we reset β_1. Thus from now on, we assume that no element $v < u_1$ ever enters or leaves U_1, which is true after finite injury.

Remark on counters: We insert at this point a remark on counters. As appears from the discussion below, we need that $\theta(i_1)$ be defined, i.e., $i_1 \in \Theta_1^C$. This is not too ambitious to require, since A is a Π_1^0 set, and thus we may assume that $i_1 \in A$. However, i_1 might have been already chosen as a witness and extracted by some $\mathcal{R}$-requirement. To avoid this, we define a **counter function** ρ, i.e., a strictly increasing computable bijection of ω onto some computable set R, such that $R \subseteq A$, and no witness for any $\mathcal{R}$-requirement is ever chosen from R, nor is of the form $x+r$ where x is some other witness and $r \in R$. Under the hypothesis that $A = \Theta^C$, we may therefore assume that $\Theta(r) = 1$ and $\Theta^C(x+r) = 1$, for every $r \in R$ and witness x. Since we need that $\Theta(i) = 1$ and $\Theta(x+i) = 1$ for every counter i and witness x —these assumptions will be used in the formal construction and in the combinatorial argument leading to the verification of Lemma 5.2— we will agree, abusing notation, that the counter i denotes in fact the i-th element of R, i.e., $\rho(i)$. Consequently, incrementing the counter i by one will mean going from $\rho(i)$ to $\rho(i+1)$.

At stages at which the $\mathcal{R}_0$-destroyer strategy β_0 passes 2., the $\mathcal{R}_1$-destroyer strategy β_1 proceeds as follows.

1. Wait for
   (a) $\Theta_1^C(x_1) = 1$;
   (b) $(\forall z \leq i_1)(z \in A \Rightarrow \Theta_1^C(z) = 1)$;
   (c) $|B^1_{i_0}| > i_1 \cdot \theta_1(i_1)$;
   (d) $U_j \upharpoonright (\theta_1(i_1)+1) = U_{j,s^*} \upharpoonright (\theta_1(i_1)+1)$ for $j = 0,1$. (In fact $\subseteq$ would be sufficient here.)

   Here $B^1_{i_0}$ is the $\gamma_1(w_0)$-use block just having been used by β_0 to destroy $\Gamma_1^{CU_1}(w_0)$, and s^* is the stage at which the use block $B^1_{i_0-1}$ was defined. The purpose of the last clause is to ensure that extracting from and putting back numbers into C after a particular stage s_* (as defined in the description of an $\mathcal{R}_3$-controller strategy below) will not reinstall Γ-axioms defined before stage s^*. The reason is that there is a permanent change in C made at stage s^*, i.e., the extraction of the i_0-1-th $\gamma_1(w_0)$ from C. Again, the second clause is to help the $\mathcal{R}$-controllers. The third clause is to ensure that $B^1_{i_0}$ is large enough for future destroying and restoring. In particular, this clause ensures that $\gamma_1(w_0) \gg \theta_1(i_1)$ (i.e., $\gamma_1(w_0)$ "much greater than" $\theta_1(i_1)$).
2. If $\min[\gamma_0(w_1) - \text{use block}] > \theta_1(x_1)$, and for all $v \geq u_1$ such that $\Delta_1^C(v) = 1$ is defined, the least element in the $\delta_1(v)$-use block is greater than $\theta_1(x_1)$, then extract x_1 from A, restrain $C \upharpoonright (\theta_1(x_1)+1)$, and stop.
3. Otherwise, define $B^0_{i_1}$ to be the current $\gamma_0(w_1)$-use block. For any $v \geq u_1$ such that $\Delta_1^C(v) = 1$ is defined, let $D^1_{i_1}(v)$ be the $\delta_1(v)$-use block. Extract $\gamma_0(w_1)$ and $\delta_1(v)$ (for each v described above) from C; request that the new $\gamma_0(w_1)$- and $\delta_1(v')$-use blocks (for any v') be very big; for any $i \leq i_1$ if $i \in U_0$ then define $\Delta_0^C(i) = U_0(i)$ with big use $\delta_0(i)$, keep it correct from

now on (unless stopped), and go back to 1. The picture is now that the uses $\gamma_0(w_1)$ and $\delta_1(v)$ all go to infinity and leave only the markers $\delta_0(i)$ for more and more i's. In fact, the δ_0-markers wipe out all lower priority markers.

The outcomes of the $\mathcal{R}_1$-destroyer strategy: $\mathcal{R}_1$ has a finite outcome (denoted by 1), if it is eventually always in 1. or in 2. It has an infinite outcome 0 if it goes from 3. to 1. infinitely often.

3.3. Description of an $\mathcal{R}_2$-destroyer strategy. Above the finite outcome of β_1, there is an $\mathcal{R}_2$-destroyer strategy β_2 similar to the strategy $\mathcal{R}_1$ which we have just described. We now look at what happens above the infinite outcome of β_1. First notice that $\mathcal{S}_1$ is not satisfied since both $\Gamma_1^{CU_1}$ and Δ_1^C have been destroyed. We have therefore to introduce another version of the $\mathcal{S}_1$-strategy, $\widehat{\mathcal{S}}_1$, which builds $\widehat{\Gamma}$. The $\mathcal{R}_2$-destroyer strategy has to deal with $\widehat{\Gamma}_1^{CU_1}$ and Δ_0^C. It will destroy $\widehat{\Gamma}_1^{CU_1}$ and build $\widehat{\Delta}_1^C$ unless it happens to find a Θ_2-computation cleared of both δ_0- and $\widehat{\gamma}_1$-uses.

The $\mathcal{R}_2$-destroyer strategy β_2 has the following parameters: A fresh witness x_2 for Θ_2; a killing point $w_2 > w_1$ from K; a threshold value u_2; and a counter i_2 to record the number of previous $\widehat{\Gamma}_1$-killings performed by β_2.

At stages at which the $\mathcal{R}_1$-destroyer strategy passes 2., the strategy β_2 proceeds as follows.

1. Wait for
   (a) $\Theta_2^C(x_2) = 1$;
   (b) $(\forall z \leq i_2)(z \in A \Rightarrow \Theta_2^C(z) = 1)$;
   (c) $|B_{i_0}^1|, |B_{i_1}^0|, |D_{i_1}^1(v)| > i_2 \cdot \theta_2(i_2)$ $(v \geq u_1)$;
   (d) $U_j \upharpoonright (\theta_2(i_2)+1) = U_{j,s^*} \upharpoonright (\theta_2(i_2)+1)$ for $j = 0, 1$.

   Here $B_{i_0}^1, B_{i_1}^0$ and $D_{i_1}^1(v)$ are the current $\gamma_1(w_0)$-, $\gamma_0(w_1)$- and $\delta_1(v)$-use blocks, respectively, and s^* is the stage at which the use block $B_{i_1-1}^0$ was defined.
2. If $\min[\widehat{\gamma}_1(w_2) - \text{use block}] > \theta_2(x_2)$, and for all $v \geq u_2$ for which $\Delta_0^C(v) = 1$ we have that $\min[\delta_0(v)\text{-use block}] > \theta_2(x_2)$, then extract x_2 from A, restrain $C \upharpoonright (\theta_2(x_2)+1)$, and stop.
3. Otherwise, define $\widehat{B}_{i_2}^1$ to be the current $\widehat{\gamma}_1(w_2)$-use block; extract $\widehat{\gamma}_1(w_2)$ from C; request that the new $\widehat{\gamma}_1(w_2)$-use block be very big; for any $i \leq i_2$ if $i \in U_1$ then define $\widehat{\Delta}_1^C(i) = U_1(i)$ with big use $\widehat{\delta}_1(i)$, keep it correct from now on (unless stopped), and go back to 1.

The outcomes of the $\mathcal{R}_2$-destroyer strategy: $\mathcal{R}_2$ has a finite outcome (denoted by 1), if it is eventually always in 1. or in 2. It has an infinite outcome 0 if it goes from 3. to 1. infinitely often.

3.4. DESCRIPTION OF AN $\mathcal{R}_3$-CONTROLLER STRATEGY. Above the infinite outcome 0 of β_2, there is an $\mathcal{R}_3$-controller strategy γ. It is a controller, since it has no Γ to kill and deals with only Δ- (in our case Δ_0- and $\widehat{\Delta}_1$-) uses. The strategy γ will eventually take charge of $\mathcal{R}_0$, $\mathcal{R}_1$ and $\mathcal{R}_2$ unless it successfully wins its own diagonalization against Θ_3. Once it takes charge, it will ensure that one of $\mathcal{R}_n$ ($n = 0, 1, 2, 3$) succeeds in diagonalizing against Θ_n^C using γ's witness.

Exactly which $\mathcal{R}_n$ wins depends on elements leaving U_0 and U_1 and the consequent effect of these extractions on the e-operators. Roughly speaking, if no elements leave either U_0 or U_1, then we do not need to correct Δ_0 and $\widehat{\Delta}_1$ for the sake of these elements. Note that we did not ignore the possibility of some new element z entering U_0 (or U_1), but to cope with this we can simply define $\Delta_0^C(z) = 1$ (or $\widehat{\Delta}_1^C(z) = 1$) with proper use. If exactly one among U_0 and U_1, respectively, has some element leaving the set, then this will clear the Θ-computation of the $\mathcal{R}_1$- or $\mathcal{R}_2$-destroyer strategy of γ_0- or $\widehat{\gamma}_1$-uses, respectively. And if both U_0 and U_1 have elements leaving, then the $\mathcal{R}_0$-destroyer strategy has a cleared Θ-computation.

The $\mathcal{R}_3$-controller strategy has a witness x_3 (the same for all Θ_n, $n = 0, 1, 2, 3$). Initially, it chooses x_3 fresh. At stages at which the $\mathcal{R}_2$-destroyer strategy β_2 passes 2., it proceeds as follows.

1. Wait for
   (a) $\Theta_n^C(x_3) = 1$ for $n = 0, 1, 2, 3$;
   (b) $i_0, i_1, i_2 > x_3$;
   (c) $\min\{|B_{i_0}^1|, |B_{i_1}^0|, |D_{i_1}^1(v)|, |\widehat{B}_{i_2}^1| : v \geq u_1\}$ to be sufficiently large (the precise bound for the size of blocks is not crucial at this moment. For example, we can choose the bound to be $4 \cdot (\theta_3(x_3) + 1) + 2$ which is larger than all possible combined changes of $U_0 \restriction (\theta_3(x_3) + 1)$ and $U_1 \restriction (\theta_3(x_3) + 1)$);
   (d) $U_j \restriction (\theta_3(x_3) + 1) = U_{j,s^*} \restriction (\theta_3(x_3) + 1)$ for $j = 0, 1$ at some stage s_*.

   Here $\widehat{B}_{i_2}^1$ is the $\widehat{\gamma}_1(w_2)$-use block currently used by β_2; and s^* is the stage at which the use block $\widehat{B}_{i_2-1}^1$ was defined. We can summarize part of the above data as follows:
   - First of all, notice that $i_0, i_1, i_2 > x_3 > x_2 > x_1 > x_0$.
   - $\widehat{\gamma}_1(w_2) > \min \widehat{B}_{i_2}^1$; by our convention, $\min \widehat{B}_{i_2}^1 > |\widehat{B}_{i_2}^1|$ and $|\widehat{B}_{i_2}^1| \gg \theta_3(x_3)$.

- $\gamma_0(w_1), \delta_1(v) \gg \theta_3(x_3)$ by the same reason. Moreover, by one of the clauses in $\mathcal{R}_2$, both $\gamma_0(w_1)$ and $\delta_1(v)$ are $\gg \theta_2(i_2)$, thus in particular, $\gg \theta_2(x_3)$ and $\gg \theta_2(x_2)$.
- $\gamma_1(w_0) \gg \theta_3(x_3), \theta_2(x_3)$ by the same reasons. Moreover, by one of the clauses in $\mathcal{R}_1$, $\gamma_1(w_0) \gg \theta_1(i_1)$, thus $\gamma_1(w_0) \gg \theta_1(x_3), \theta_1(x_1)$.

2. Set
$$y_n = \max_{n \le m \le 3} \{\theta_m(x_3)\}$$
for $n = 0, 1, 2, 3$; set $i_n^* = i_n$ for $n = 0, 1, 2$; extract x_3 from A, and restrain $C \restriction (y_0 + 1)$ from lower priority strategies. We say that the $\mathcal{R}_3$-controller strategy γ **is taking charge of** the β_0-, β_1-, and β_2-strategy. (Some of the above relations about uses can be restated in terms of the y's as follows: We have $\widehat{\gamma}_1(w_2) \gg y_3$; $\gamma_0(w_1), \delta_1(v) \gg y_2$; $\gamma_1(w_0) \gg y_1$.)
3. From now on, whenever some element below y_0 leaves U_j, go through the following algorithm to decide which strategies should act, and act accordingly.

   Find the unique strategy which kills Γ_0 for $\mathcal{S}_0$. In our case, it is the $\mathcal{R}_1$-destroyer strategy β_1. Find the biggest y which is definitely below $\gamma_0(w_1)$. In our case it is y_2. Ask if
$$U_0 \restriction (y_2 + 1) \supseteq U_{0,s_*} \restriction (y_2 + 1),$$
   i.e., if no elements have left $U_{0,s_*} \restriction (y_2 + 1)$.

   a. The answer is "Yes". It means that y_2 is safe from Δ_0^C-corrections, i.e., we will not extract any number less than y_2 from C (since $\delta_0(v) > y_2$ for every $v > y_2$) to correct $\Delta_0^C(z)$ for some z. Thus the $\mathcal{R}_1$-destroyer strategy β_1 can continue to act. We then repeat the same procedure for the strategies respecting Δ_1 (this makes us move up along the tree). In our case, we restrict our attention to nodes above $\beta_1\hat{\ }0$; by repeating the algorithm, we find the next higher $\mathcal{S}$-strategy whose Γ was destroyed (in our case $\widehat{\mathcal{S}}_1$) and a number, which is definitely below $\widehat{\gamma}_1(w_2)$ (in our case y_3). Ask if
$$U_1 \restriction (y_3 + 1) \supseteq U_{1,s_*} \restriction (y_3 + 1).$$

   a.1. The answer is "Yes". This means that y_3 is safe from $\widehat{\Delta}_1$-corrections. Thus, the requirement $\mathcal{R}_3$ is satisfied, and Δ_0^C and $\widehat{\Delta}_1^C$ are correct. We let the strategies β_n act for $n = 0, 1, 2$; and have $C \cap B \not\supseteq C_{s_*} \cap B$ (by extracting elements from B) for

$B = B^1_{i_0^*}$, $B^0_{i_1^*}$, $D^1_{i_1^*}(v)$, $\widehat{B}^1_{i_2^*}$. The purpose is to kill the definitions made at stage s_*.

a.2 The answer is "No". This means that $\widehat{\gamma}_1(w_2)$ is cleared, since $y_3 < \widehat{\gamma}_1(w_2)$. Thus, the requirement $\mathcal{R}_2$ is satisfied via $\Theta_2^C(x_3) \neq A(x_3)$ and the computation is cleared of $\widehat{\Gamma}_1$-uses and Δ_0 is correct. We let the strategies β_n act for $n = 0, 1$; prevent the $\mathcal{R}_2$-destroyer strategy β_2 from acting; restore $C \upharpoonright (y_2 + 1) \supseteq C_{s_*} \upharpoonright (y_2 + 1)$; and have $C \cap B \not\supseteq C_{s_*} \cap B$ for $B = B^1_{i_0^*}$, $B^0_{i_1^*}$, $D^1_{i_1^*}(v)$.

b. The answer is "No". This means that $\gamma_0(w_1)$ is cleared, since $y_2 < \gamma_1(w_1)$. We switch back to Γ_0 (i.e., stop building Δ_0). We repeat the same procedure for the strategies having higher priority (which makes us move down along the tree). In our case, similar to a. we find $\mathcal{S}_1$ and y_1. Ask if $U_1 \upharpoonright (y_1 + 1) \supseteq U_{1,s_*} \upharpoonright (y_1 + 1)$.

b.1. The answer is "Yes". This means that the computation $\Theta_1^C(x_3)$ is safe from Δ_1-corrections and cleared of Γ_0-uses (by U_0-change). Therefore, the requirement $\mathcal{R}_1$ is satisfied via the witness x_3. We let the $\mathcal{R}_0$-destroyer strategy β_0 act and prevent the $\mathcal{R}_n$-strategies from acting for $n = 1, 2$; restore $C \upharpoonright (y_1 + 1) \supseteq C_{s_*} \upharpoonright (y_1 + 1)$; and have $C \cap B \not\supseteq C_{s_*} \cap B$ for $B = B^1_{i_0^*}$.

b.2. The answer is "No". This means that $\gamma_1(w_0)$ is cleared, since $y_1 < \gamma_1(w_0)$. Thus, $\mathcal{R}_0$ is satisfied via the γ_1- and γ_0-cleared computation $\Theta_0^C(x_3)$. We prevent the $\mathcal{R}_n$-destroyer strategies β_n from acting for $n = 0, 1, 2$; and restore $C \upharpoonright (y_0 + 1) \supseteq C_{s_*} \upharpoonright (y_0 + 1)$.

In any case, one of the requirements $\mathcal{R}_n$ (for $n = 0, 1, 2, 3$) is satisfied.

4. The construction

4.1. The tree of strategies. We use a tree T which is a subtree of the full binary tree $2^{<\omega}$ to organize our strategies. We interpret 0 as the infinite outcome and 1 as the finite outcome of a strategy $\xi \in T$.

A remark on notations: In the following the Greek letter ξ is reserved for a generic node of the tree, α for an $\mathcal{S}$-strategy, and β for an $\mathcal{R}$-destroyer strategy and γ for an $\mathcal{R}$-controller strategy.

Fix an effective priority ranking of the requirements:

$$\mathcal{S}_0 < \mathcal{R}_0 < \mathcal{S}_1 < \mathcal{R}_1 < \cdots.$$

We label each node on T with a strategy recursively as follows (in fact we are defining T simultaneously by the same recursion):

- the root $\emptyset$ is labelled by $\mathcal{S}_0$:
- let $\xi \in T$. Assume that all $\eta \subset \xi$ have been labelled. We first introduce some terminology.

  We say that a strategy $\eta \subset \xi$ is **Σ_3-injured at ξ by the pair α and β**, if α is an $\mathcal{S}$-strategy, β an $\mathcal{R}$-destroyer strategy,

$$\alpha \subset \eta \subset \beta \subset \beta\hat{\ }0 \subseteq \xi$$

  and β is targeted to destroy α's Γ (as defined below). We say that $\eta \subset \xi$ is **Σ_3-injured at ξ**, if there is a pair α and β which Σ_3-injures η at ξ. Note that by the assignment procedure, η can be either an $\mathcal{S}$- or an $\mathcal{R}$-destroyer strategy but not an $\mathcal{R}$-controller, because no $\mathcal{S}$ will be active (as defined below) at an $\mathcal{R}$-controller.

  A requirement $\mathcal{S}$ is said to be **active at ξ** if there is an $\mathcal{S}$-strategy $\alpha \subset \xi$ and there is no $\mathcal{R}$-destroyer strategy β with $\alpha \subset \beta\hat{\ }0 \subseteq \xi$ that is targeted to destroy α's Γ (as defined below).

  A requirement $\mathcal{S}$ is said to be **satisfied at ξ** if there is an $\mathcal{R}$-destroyer strategy β such that $\beta\hat{\ }0 \subseteq \xi$, β is targeted to destroy the Γ of an $\mathcal{S}$-strategy at $\alpha \subset \beta$ and neither α nor β is Σ_3-injured at ξ. We will say that α is satisfied at ξ **via** β. Observe that in the above setting, α is Σ_3-injured if and only if β is.

  A requirement $\mathcal{R}$ is **satisfied at** ξ if there is an $\mathcal{R}$-destroyer or a $\mathcal{R}$-controller strategy η with $\eta\hat{\ }1 \subseteq \xi$.

  We now continue the labelling process of ξ. The node ξ is labelled with the strategy of highest-priority requirement that is neither active nor satisfied at ξ.

  If ξ is labelled with an $\mathcal{R}$-strategy then ξ is an **$\mathcal{R}$-controller strategy** if no requirement $\mathcal{S}$ is active at ξ; otherwise ξ is an **$\mathcal{R}$-destroyer strategy** targeted to destroy the Γ of the longest $\alpha \subset \xi$ such that α's $\mathcal{S}$-requirement is active at ξ.

  The **immediate successors** of ξ on T are $\xi\hat{\ }0$ if ξ is an $\mathcal{S}$-strategy, $\xi\hat{\ }1$ if ξ is an $\mathcal{R}$-controller strategy, and both $\xi\hat{\ }0$ and $\xi\hat{\ }1$ if ξ is an $\mathcal{R}$-destroyer strategy.

Notice that our labelling process guarantees the following fact: If α is active at ξ then α is not Σ_3-injured at ξ.

LEMMA 4.1 (Finite Injury and Satisfaction along Any Path Lemma): *Assume* *that p is a path through T and $\mathcal{O}$ is a requirement. Then $\mathcal{O}$ is assigned to only finitely many nodes $\xi \subset p$. If ξ_0 is the longest such, then either $\mathcal{O}$ is satisfied at $p \upharpoonright n$ via ξ_0 for all $n > |\xi_0|$ or $\mathcal{O}$ is active at $p \upharpoonright n$ via ξ_0 for all $n > |\xi_0|$.*

Proof: Routine. See for instance [7]. ∎

4.2. Decision Algorithm. We now develop an algorithm which will be used in the description of the construction and the verification. Fix an arbitrary $\tilde{R}$-controller strategy γ. Let

$$\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{j_0}$$

be the $\mathcal{S}$-requirements of higher priority than $\tilde{R}$. Let

$$\alpha_0 \subset \alpha_1 \subset \cdots \subset \alpha_{k_0}$$

be the $\mathcal{S}$-strategies $\alpha \subset \gamma$; let

$$\beta_0 \subset \beta_1 \subset \cdots \subset \beta_{l_0 - 1}$$

be the $\mathcal{R}$-destroyer strategies β with $\beta\hat{\ }0 \subset \gamma$ and let β_{l_0} be γ. We shall call the set $\{\beta_l : 0 \leq l \leq l_0\}$ the *domain of* γ.

Let U_j be the 3-c.e. set corresponding to the requirement $\mathcal{S}_j$ for $0 \leq j \leq j_0$. Fix a stage s_* and numbers $y_0 \geq y_1 \geq \ldots \geq y_{l_0}$. At any stage $s > s_*$, we may use the following algorithm, called the **decision algorithm for** γ, to find a unique number l_* with $0 \leq l_* \leq l_0$. The decision made will be based on the assignment of strategies to nodes of the tree T and on the elements leaving the sets U_j between stages s_* and s. The number l_* will be used in the sequel to partition the domain of γ into two sets: $\{\beta_l : l \geq l_*\}$ in which every node is stopped by γ, and $\{\beta_l : l < l_*\}$ in which all nodes are allowed to act by γ.

The algorithm has two parameters a and b for the lower and upper bound of the search, respectively. We run through cycles starting from $i = 0$. After each cycle i, the searching scope is shrunk to an interval $[a_{i+1}, b_{i+1}]$ containing only the $\mathcal{S}_{i'}$-strategies for $i < i' \leq j_0$.

Roughly speaking, at each cycle i, we will decide whether or not we can make use of the fact that some element of $U_{i,s_*} \upharpoonright y_l$ leaves U_i to get a γ_i-cleared computation. The number l is related to the different $\mathcal{S}_i$-strategies working for the same $\mathcal{S}_i$-requirement. For instance, in the discussion of two $\mathcal{S}$-strategies of Section 3, the requirement $\mathcal{S}_1$ has two $\mathcal{S}_1$-strategies, namely $\mathcal{S}_1$ and $\widehat{\mathcal{S}}_1$. By observing that a new strategy appears only after the old one is Σ_3-injured, we can use the global priority of requirements as a guide in our search.

Initially set $a_0 = -1$ and $b_0 = l_0$. Let us use β_{-1} to denote some imaginary node below the root of the tree, so that we are sure to include the $\mathcal{S}_0$-strategy which is assigned to the root.

CYCLE i: Given $a = a_i$ and $b = b_i$. If there is no $\mathcal{S}_i$-strategy α such that $\beta_a \subset \alpha \subset \beta_b$, or $b = 0$, then stop and output $l_* = b$.

Otherwise, let α be the unique node labelled $\mathcal{S}_i$ such that $\beta_a \subset \alpha \subset \beta_b$ and let l be the unique number such that $a < l < b$ and the $\mathcal{R}$-destroyer strategy β_l is targeted to destroy α's Γ. (We will show the existence of α and l later.) Check if

$$U_{i,s_*} \upharpoonright (y_{l+1} + 1) \subseteq U_{i,s} \upharpoonright (y_{l+1} + 1).$$

If the answer is "Yes", then set $a_{i+1} = l$ and $b_{i+1} = b$; otherwise, set $a_{i+1} = a$ and $b_{i+1} = l$. Go to cycle $i + 1$.

This ends the decision algorithm.

The following lemma verifies the existence of the parameters mentioned in the decision algorithm.

LEMMA 4.2 (Decision Lemma): *Let $i \leq j_0$ and $a = a_i$ and $b = b_i$. Assume that the algorithm does not stop at cycle i. Then the following holds.*

(1) *There is a unique $\mathcal{S}_i$-strategy α such that $\beta_a \subset \alpha \subset \beta_b$.*
(2) *For any $i'' > i' \geq i$, if there is an $\mathcal{S}_{i''}$-strategy α'' with $\beta_a \subset \alpha'' \subset \beta_b$, then there is an $\mathcal{S}_{i'}$-strategy α' with $\beta_a \subset \alpha' \subset \alpha'' \subset \beta_b$.*
(3) *For any $i' < i$, either* (3a) *or* (3b) *holds:*
   (3a) *The requirement $\mathcal{S}_{i'}$ is active at β_m and*

$$U_{i',s_*} \upharpoonright (y_{m+1} + 1) \not\subseteq U_{i',s} \upharpoonright (y_{m+1} + 1)$$

   for all m with $a < m \leq b$; or
   (3b) *the requirement $\mathcal{S}_{i'}$ is satisfied at β_m via β_a and*

$$U_{i',s_*} \upharpoonright (y_{m+1} + 1) \subseteq U_{i',s} \upharpoonright (y_{m+1} + 1)$$

   for all m with $a < m \leq b$.

   Consequently, no nodes between β_a and β_b are labelled with an $\mathcal{S}_{i'}$-strategy.
(4) *There is a unique l such that $a < l < b$ and the $\mathcal{R}$-destroyer strategy β_l is targeted to destroy α's Γ.*
(5) *For any m with $a < m \leq b$ and for any $i' < i$, β_m is not targeted to destroy $\mathcal{S}_{i'}$-strategy's Γ.*

Proof: We argue by induction on i.

CASE $i = 0$: Since the $\mathcal{S}_0$-strategy is uniquely assigned to the root, (1) holds. (2) follows from the labelling process of strategies. (3) and (5) hold trivially. For (4), since $\beta_b = \gamma$ is an $\mathcal{R}$-controller strategy, there is some (least) β_l with $l < l_0 = b$ which destroys $\mathcal{S}_0$'s Γ. And uniqueness follows from the fact that $\mathcal{S}_0$ is not active for all $\beta \supset \beta_l$.

CASE $i+1$: Suppose that conditions (1)–(5) hold for i. Let $a = a_i$, $b = b_i$, α be the unique $\mathcal{S}_i$-strategy with $\beta_a \subset \alpha \subset \beta_b$ and let β_l be the unique $\mathcal{R}$-destroyer strategy with $a < l < b$ which is targeted to destroy α's Γ.

We first argue that (3) holds for $i+1$. By inductive hypothesis, there are neither $\mathcal{S}_j$-strategies (for $j \geq i$) nor any $\mathcal{R}$-destroyer strategy β_m between β_a and α. Thus for all $m \leq l$, $\mathcal{S}_i$ is active via α at β_m; and for all m with $l < m \leq b$, $\mathcal{S}_i$ is satisfied via β_l at β_m. Moreover, if $U_{i,s_*} \upharpoonright (y_{l+1}+1) \not\subseteq U_{i,s} \upharpoonright (y_{l+1}+1)$, then by the algorithm, $a_{i+1} = a$ and $b_{i+1} = l$. Thus for all m with $a = a_{i+1} < m \leq b_{i+1} = l$, we have that $\mathcal{S}_i$ is active at β_m, and $U_{i,s_*} \upharpoonright (y_{m+1}+1) \not\subseteq U_{i,s} \upharpoonright (y_{m+1}+1)$ by the monotonicity of the numbers y_m, thus (3a) holds. Similarly, if $U_{i,s_*} \upharpoonright (y_{l+1}+1) \subseteq U_{i,s} \upharpoonright (y_{l+1}+1)$, then by the algorithm, $a_{i+1} = l$ and $b_{i+1} = b$. Thus for all m with $l = a_{i+1} < m \leq b_{i+1} = b$, we have that $\mathcal{S}_i$ is satisfied at β_m, and $U_{i,s_*} \upharpoonright (y_{m+1}+1) \subseteq U_{i,s} \upharpoonright (y_{m+1}+1)$ by monotonicity of the numbers y_m, thus (3b) holds.

We now prove (2) for $i+1$. Consider $i'' > i' \geq i+1$ such that there is an $\mathcal{S}_{i''}$-strategy α'' with $\beta_{a_{i+1}} \subset \alpha'' \subset \beta_{b_{i+1}}$. Then by induction hypothesis, there is an $\mathcal{S}'$-strategy α' with $\beta_a \subset \alpha' \subset \alpha'' \subset \beta_b$. The only worry is that $\alpha' \subset \beta_l \subset \alpha''$. But then α' would be Σ_3-injured by β_l, and thus there will be another $\mathcal{S}_{i'}$-strategy between β_l and β_b. Thus in any case, there is an $\mathcal{S}_{i'}$-strategy between $\beta_{a_{i+1}}$ and $\beta_{b_{i+1}}$.

We now prove (1) and (4) for $i+1$. By the fact that the algorithm does not stop at cycle $i+1$ and (2), there is an $\mathcal{S}_{i+1}$-strategy α' with $\beta_{a_{i+1}} \subset \alpha' \subset \beta_{b_{i+1}}$. To see the uniqueness of α', we look at two cases depending on the position of the $\mathcal{S}_{i+1}$-strategy α'.

CASE 1: $\beta_a \subset \alpha' \subset \beta_l$. Then pick the least such α'. There is an l' with $a < l' < l$ such that $\beta_{l'}$ is targeted to destroy α' (otherwise, β_l would not be targeted to destroy α). Now at any node extending $\beta_{l'}$, $\mathcal{S}_{i+1}$ remains to be satisfied, unless it is Σ_3-injured by some pair η and ξ where η is an $\mathcal{S}_{i'}$-strategy for some $i' < i+1$. Since $\xi \subset \beta_l$, by (5), $i' \geq i$, and by (4), $i' \neq i$, a contradiction.

CASE 2: $\beta_l \subset \alpha' \subset \beta_b$. The argument is similar, and it is left to the reader.

Finally, (5) follows from a two-case discussion similar to the one given for (4). ∎

Note that there are at most j_0 many cycles, so the algorithm terminates. We now single out an easy consequence of the Decision Lemma which establishes the desired properties of β_{l_*}.

LEMMA 4.3: *Let l_* be the number obtained from the decision algorithm for γ. Suppose that the algorithm does not stop at cycle i. Then*

(a) *the requirement $\mathcal{S}_i$ is active at β_{l_*} if and only if*

$$U_{i,s_*} \restriction (y_{l_*+1}+1) \not\subseteq U_{i,s} \restriction (y_{l_*+1}+1);$$

and

(b) *the requirement $\mathcal{S}_i$ is satisfied at β_{l_*} if and only if*

$$U_{i,s_*} \restriction (y_{l_*+1}+1) \subseteq U_{i,s} \restriction (y_{l_*+1}+1).$$

Proof: It follows from (3a) and (3b) in Lemma 4.2 by replacing m with l_*. ∎

4.3. CONSTRUCTION BY STAGES. We now describe the stage by stage construction. We construct a 3-c.e. set C, an auxiliary Π^0_1-set A, an e-operator Γ_α for each $\mathcal{S}$-strategy $\alpha \in T$, and an e-operator Δ_β for each $\mathcal{R}$-destroyer strategy $\beta \in T$.

Whenever a strategy ξ is **initialized**, its parameters become all undefined, its e-operator (if any) becomes totally undefined, and ξ no longer takes charge of any other strategy. The same holds when a strategy is **reset** except that then the killing point w and the threshold u of an $\mathcal{R}$-destroyer strategy do not become undefined.

At stage 0, all strategies are initialized, and C and A are set to be ω.

At stage $s+1$, first of all, for each $\mathcal{R}$-destroyer strategy β such that $\overline{K}_{s+1} \restriction w_\beta \neq \overline{K}_s \restriction w_\beta$ or $U_{j,s+1} \restriction u_\beta \neq U_{j,s} \restriction u_\beta$ for some $\mathcal{S}_j$-strategy $\alpha \subset \beta$, we reset the strategies $\xi \in T$ with $\beta \leq \xi$, where $\leq$ is the ordering on the tree T.

Next, we let certain strategies on T be **eligible to act** at $s+1$ as follows. First we let $\emptyset$ be eligible to act. Given ξ that is eligible to act, we may allow an immediate successor of ξ (determined by ξ's action as defined below) to be eligible to act next. When we proceed to stage $s+2$ we initialize all strategies $\eta > \xi$ if $|\xi| = s$, or initialize all strategies η with $\xi' < \eta$ if the stage is ended by some strategy ξ'.

We describe the action of an individual strategy ξ, depending on what type of strategy ξ is. (We work below with approximations by stages to various e-operators and sets involved in the construction. Then when for example in Case 1 we write $\Gamma^{CU}(w) \neq \overline{K}(w)$ we should in fact write $\Gamma^{CU}[s](w) \neq \overline{K}[s](w)$. Unless strictly necessary, for notational simplicity we omit to append $[s]$.)

CASE 1: ξ is an $\mathcal{S}$-strategy.

Find the least $w \le s$ such that $\Gamma^{CU}(w) \neq \overline{K}(w)$. If no such w exists, then do nothing. Otherwise, proceed according to the subcase that applies:

CASE 1a: $\Gamma^{CU}(w) = 1$ and $w \in K$. Extract the least unused element of the current $\gamma(w)$-use block from C.

CASE 1b: $w \in \overline{K}$ and $\Gamma^{CU}(w) = 1$ has been defined before but for any t with $s' < t < s+1$ and for any $w' \le w$, $C_t \upharpoonright B(w') = C_{s'} \upharpoonright B(w')$, where s' is the maximal stage such that $\Gamma^{CU}(w)[s'] = 1$. Redefine $\Gamma^{CU}(w) = 1$ with the largest $\gamma(w)$-use block defined so far as the new $\gamma(w)$-use block.

CASE 1c: Otherwise, we (re)define $\Gamma^{CU}(w) = 1$ with a big use.

In any case, $\xi\hat{\ }0$ is eligible to act next.

CASE 2: ξ is an $\mathcal{R}$-destroyer strategy targeted to destroy the Γ_α of some $\mathcal{S}$-strategy $\alpha \subset \xi$.

Let $i = i_\xi$. First check if ξ's killing point $w = w_\xi$ or threshold $u = u_\xi$ or ξ's witness $x = x_\xi$ is undefined. If so then redefine it/them fresh, and let $i = 0$. (See also the convention on counters made in Section 3.)

Next check if ξ has stopped by itself (as defined below) since it was last initialized or reset. If so, then do nothing and let $\xi\hat{\ }1$ be eligible to act next.

Next check if one or more $\mathcal{R}$-controller strategies $\gamma \supset \xi$ are taking charge of ξ (as defined below). If so, then let these γ's act now in decreasing order of priority (with respect to the ordering $\le$ on T) according to Case 3b below. If one of these γ's stops ξ then do nothing and let $\xi\hat{\ }1$ be eligible to act next.

Finally, check whether or not the following conditions (1) to (5) hold:

$$\Theta^C(x) = 1, \tag{1}$$

$$(\forall z \le x+i+1)(z \in A \Rightarrow \Theta^C(z) = 1), \tag{2}$$

$$(\forall \beta \in \mathcal{D})\ (i_\beta > \theta(x+i)), \tag{3}$$

where $\mathcal{D}$ is the set of all $\mathcal{R}$-destroyer strategies β with $\beta\hat{\ }0 \subseteq \xi$;

$$(\forall B \in \mathcal{B}_0)\ (|B| > i \cdot (\theta(x+i)+1)), \tag{4}$$

where $\mathcal{B}_0$ is the set of all use blocks used by some $\beta \in \mathcal{D}$ to destroy an e-operator at the current stage. (Since the blocks used by β to destroy will not become clear until later, let us add a few more words of explanation. As we will see, each $\beta \in \mathcal{D}$ is targeted to destroy $\Gamma_{\alpha'}$ for some $\alpha' \subset \beta$ and a few $\Delta_{\overline{\beta}}$, for $\alpha' \subset \overline{\beta} \subset \beta$.

For each such $\beta, \overline{\beta}$, $\mathcal{B}_0$ contains the $\gamma_{\alpha'}(w_\beta)$-use block and the $\delta_{\overline{\beta}}(v_{\overline{\beta}})$-use block, where $v_{\overline{\beta}}$ is the number $\geq u_\beta$ with the least $\delta_{\overline{\beta}}$-use.)

And

$$(5)\qquad \begin{gathered}(\forall \mathcal{S}\text{-strategies } \alpha' \subset \beta)(\forall s')(s^* \leq s' \leq s+1 \Rightarrow \\ U_{\alpha,s'} \upharpoonright (\theta(x+i)+1) = U_{\alpha,s^*} \upharpoonright (\theta(x+i)+1))\end{gathered}$$

where

$$s^* = \max\{s' \leq s : s' = 0 \text{ or } \xi \text{ was eligible to act at stage } s'\}.$$

If neither condition applies then do nothing and let $\xi\hat{\ }1$ be eligible to act next. Otherwise, let $\mathcal{B}_1$ be the collection of

- all current $\gamma_{\alpha'}(w)$-use blocks B such that some requirement $\mathcal{S}$ is active at ξ via α';
- all $\delta_\beta(v)$-use blocks D such that some requirement $\mathcal{S}$ is satisfied at ξ via β, with $v \geq u$, and $\Delta^C_\beta(v) = 1$ is defined.

Check if

$$(6)\qquad (\forall B \in \mathcal{B}_1)\ (|B| > \theta(x)).$$

If yes, then we declare that ξ **stops by itself**, extract x from A, and let ξ end the stage.

Otherwise, we destroy the e-operators Γ_α and Δ_β where β has lower priority than α as follows: Extract the least unused element of the $\gamma_\alpha(w)$-use block from C; and for each $\mathcal{R}$-destroyer strategy β with $\alpha \subset \beta\hat{\ }0 \subseteq \xi$, find $v \geq u$ such that $\Delta_\beta(v) = 1$ is defined and has the least δ_β-use, extract the least unused element of the $\delta_\beta(v)$-use block from C.

We then increment the counter i by one, and update the e-operator Δ^C as follows: Find the least $z < i+1$ (if any) such that $\Delta^C(z) \neq U(z)$. If $z \notin U$ then extract the least element of the $\Delta^C(z)$-use block from C. If $z \in U$ and no element left the use block $B(z)$, then define $\Delta^C(z) = 1$ with the previous use. Otherwise, define $\Delta^C(z) = 1$ with big use.

End ξ's action at this stage by letting $\xi\hat{\ }0$ be eligible to act next.

Case 3: ξ is an $\mathcal{R}$-controller strategy. Let

$$\beta_0 \subset \beta_1 \subset \cdots \subset \beta_{l_0-1}$$

be the $\mathcal{R}$-destroyer strategies β with $\beta\hat{\ }0 \subseteq \xi$ and let β_{l_0} be ξ. Let j_0 be the number of $\mathcal{S}$-requirements whose strategies are assigned at nodes $\alpha \subset \xi$.

CASE 3a: ξ is currently not taking charge of other strategies. First, check if ξ's witness x is undefined. If so redefine it fresh.

Next, denote by $i_0, \ldots, i_{l_0-1}$ the parameters i of $\beta_0, \ldots, \beta_{l_0-1}$, respectively. Let g be a computable function to be determined later. Let $\mathcal{B}_2$ be the set of all use blocks used by some β_l (for $0 \le l < l_0$) to destroy an e-operator at the current stage. Check if

$$(\forall l < l_0)\ (i_l > \max\{x, g(j_0)\}), \tag{7}$$

$$\Theta^C(x) = 1, \tag{8}$$

$$(\forall B \in \mathcal{B}_2)\ (|B| > g(j_0) \cdot (\theta(x) + 1) + 1)), \tag{9}$$

and

$$\begin{gathered}(\forall \mathcal{S}\text{-strategies } \alpha \subset \xi)(\forall s')(s^* \le s' \le s+1 \Rightarrow \\ U_{\alpha,s'} \restriction (\theta(x)+1) = U_{\alpha,s^*} \restriction (\theta(x)+1))\end{gathered} \tag{10}$$

and $s^* = \max\{s' \le s : s' = 0 \text{ or } \xi \text{ was eligible to act at stage } s'\}$.

If none applies then end ξ's action at this stage by letting $\xi\hat{\ }1$ be eligible to act next. Otherwise, say that ξ is **taking charge of** $\beta_0, \ldots, \beta_{l_0-1}$; extract x from A; set $y_l = \max\{\theta_{\beta_{l'}}(x) : l \le l' \le l_0\}$, for $l \le l_0$; set $s_* = s$; say ξ does not stop any of $\beta_0, \ldots, \beta_{l_0-1}$; set $l_* = l_0$; and end the stage.

CASE 3b: ξ is currently taking charge of $\beta_0, \ldots, \beta_{l_0-1}$. Apply the decision algorithm for ξ to find l_* such that ξ **stops** β_l for $l_* \le l < l_0$, and ξ **does not stop** β_l for $0 \le l < l_*$. Let $\mathcal{B}_3$ be the set of all use blocks used by some β_l (for $0 \le l < l_*$) to destroy an e-operator at stage s_*. Now restore

$$C \restriction (y_{l_*}+1) \supseteq C_{s_*} \restriction (y_{l_*}+1) \tag{11}$$

by possibly putting back elements into C, and ensure

$$(\forall B \in \mathcal{B}_3)\ (C \cap B \not\supseteq C_{s_*} \cap B) \tag{12}$$

by possibly extracting the least unused element from C. (Lemma 5.2 will guarantee that this is always possible.)

Furthermore, if l_* has changed since the last time when ξ (re)set l_* then ξ ends the stage. Otherwise, if some $\tilde{R}$-destroyer strategy β has let ξ act first then return to β's action; otherwise, end ξ's action at this stage by letting $\xi\hat{\ }1$ be eligible to act next.

This ends the construction.

5. Verification

We begin the verification with the usual definition of true path. The **true path** f is the path through T defined inductively as follows. Suppose $\xi = f \upharpoonright n$. Then:

(i) $f(n) = 0$ if ξ is an $\mathcal{S}$-strategy.

(ii) $f(n) = 1$ if ξ is an $\mathcal{R}$-controller strategy.

(iii) $f(n) = 0$ if ξ is an $\mathcal{R}$-destroyer strategy and $\xi\hat{\ }0$ is eligible to act at infinitely many stages; otherwise $f(n) = 1$.

LEMMA 5.1 (Finite Initialization Along the True Path Lemma): *Any strategy $\xi \subset f$ is initialized or reset at most finitely often, and it is eligible to act at infinitely many stages.*

Proof: Routine. See, e.g., [7]. ∎

Next we do a counting argument to show that the size of the use blocks is sufficiently large to carry out the construction. More precisely, let $\gamma = \beta_{l_0}$ be a fixed $\mathcal{R}$-controller strategy and let the domain of γ be $\{\beta_0, \beta_1, \ldots, \beta_{l_0-1}\}$. Suppose that at stage s_*, γ takes charge of β_l for $0 \le l < l_0$.

LEMMA 5.2: *The following hold:*

(a) *For any $z \in \omega$, if z has ever been extracted from C, then there is at most one $\mathcal{R}$-controller strategy that can put z back into C. If z has been put back, then it will never be extracted again. Consequently, C is a 3-c.e. set.*

(b) *There is a computable function g such that: If (3) and (4) are obeyed by every β_l for $0 \le l < l_0$, and (7) and (9) are obeyed by γ and g, then for any stage $s > s_*$ and for any use block B used by some β_l, $0 \le l < l_*$, there is an unused element in $C \cap B$, where l_* is the number obtained by executing the decision algorithm for γ at stage s. In other words, the size of B is larger than the number of times for γ to execute (11) and (12).*

(c) *For any $n \ge 3$ or $n = \omega$, there is a computable function g_n such that (b) holds where the 3-c.e. sets U_j in the requirements are replaced by n-c.e. sets and g is replaced by g_n.*

Proof: Let us consider a number z which is in some use block B and it is extracted from C at some stage s_z. Suppose that at some stage s^+, z is put back into C. Since only an $\mathcal{R}$-controller strategy can put elements back into C, let us assume that γ is the one which puts z back at the minimal stage s^+. It must be that γ wants to restore $C \cap B$ back to some stage s_* such that $s_* < s_z < s^+$. As different strategies work on different use blocks, only some $\mathcal{R}$-controller strategy γ' might want to extract z from C again, to restore C back to some stage s'_*

when z is out, that is $s_z \leq s'_* < s^+$. If $\gamma' > \gamma$, then γ' gets initialized at stage s_*, hence it will work on a different block; if $\gamma' < \gamma$ then at stage s'_*, γ' would initialize γ, contradicting the choice of s^+. This establishes (a).

We now count the number of changes in B at β_l, with $l < l_*$. By the argument in (a), we need only consider the changes made by one $\mathcal{R}$-controller strategy γ. Assume that γ never gets initialized after stage s_*. We now trace the change in B back to some U_j-change on some element less than y_{l+1}, and we show that $l < l_*$ only if there exists $j \leq j_0$ such that $U_j \upharpoonright y_{l+1} \not\supseteq U_{j,s_*} \upharpoonright y_{l+1}$, or $l_* = l_0$. Suppose that $l_* \neq l_0$. Then there is a cycle j in the decision algorithm such that $b_{j+1} \neq b_j$, i.e., there is a k such that $U_j \upharpoonright y_{k+1} \not\supseteq U_{j,s_*} \upharpoonright y_{k+1}$ and $b_{j+1} = k$. Now $l < l_*$ so $l \leq b_{j+1} = k$. Thus $y_{l+1} \geq y_{k+1}$, hence $U_j \upharpoonright y_{l+1} \not\supseteq U_{j,s_*} \upharpoonright y_{l+1}$.

For a fixed j, since U_j is 3-c.e., $U_j \upharpoonright y_{l+1}$ can change at most $3(y_{l+1}+1)$ many times, i.e., $3(j_0+1)(y_{l+1}+1)$ many times for all $j \leq j_0$ combined (in fact, as far as our construction is concerned, only the elements that leave U_j matter). Let $g(j) = 3(j+1)$. Then γ will extract at most $g(j_0) \cdot (y_{l+1}+1)$ many numbers from the use block B at β_l.

We now check the size of B used by β_l. If $l+1 = l_0$, then by (9), $|B| > g(j_0) \cdot (\theta_\gamma(x_\gamma)+1)+1$, and we are done.

For $l+1 < l_0$, we first observe that

$$\begin{aligned} \theta_{\beta_{l+1}}(i_{\beta_{l+1}}) \geq & i_{\beta_{l+1}} \\ > & \theta_{\beta_{l'}}(i_{\beta_{l'}}) \text{ by (3) for } l' > l+1 \\ \geq & \theta_{\beta_{l'}}(x_\gamma) \text{ by (7) and convention on } \Theta\text{-uses,} \end{aligned}$$

which implies

$$\theta_{\beta_{l+1}}(i_{\beta_{l+1}}) \geq y_{l+1} = \max\{\theta_{\beta_{l'}}(x_\gamma) : l < l' \leq l_0\}.$$

Now by (4),

$$\begin{aligned} |B| > & i_{\beta_{l+1}} \cdot (\theta_{\beta_{l+1}}(i_{\beta_{l+1}})+1) \\ \geq & g(j_0) \cdot (\theta_{\beta_{l+1}}(i_{\beta_{l+1}})+1) \text{ by (7)} \\ \geq & g(j_0)(y_{l+1}+1)+1, \end{aligned}$$

which shows that B is large enough and establishes (b).

For (c), let $g_n(j) = n \cdot (j+1)$ for the n-c.e. case where $n \geq 3$, and let $g(x,j) = (j+1)x$ for the ω-c.e. case. The rest is similar to the argument in (b). ∎

Lemma 5.3: *Every requirement $\mathcal{S}$ is satisfied.*

Proof: Fix $\mathcal{S}$. By Lemma 4.1, there is a unique $\mathcal{S}$-strategy $\alpha \subset f$ such that the requirement $\mathcal{S}$ is either active at $f \upharpoonright n$ via α for all $n > |\alpha|$, or satisfied at

$f \upharpoonright n$ via some $\mathcal{R}$-destroyer strategy β for all $n > |\beta|$. Since α (or β, respectively) is initialized or reset only finitely often, there is a stage s_0 after which α (or β, respectively) never gets initialized or reset, hence α (or β) will work on a fixed e-operator Γ_α (or Δ_β respectively).

CASE 1: $\mathcal{S}$ is active at $f \upharpoonright n$ for all $n > |\alpha|$.

We will show that $\overline{K} = \Gamma^{CU}$, where $\Gamma = \Gamma_\alpha$.

Suppose that $\overline{K} \neq \Gamma^{CU}$. Let x be the least counterexample. Let $s_1 \geq s_0$ be a stage such that

$$(\forall t > s_1)(\overline{K}[t] \upharpoonright (x+1) = \overline{K}[s_1] \upharpoonright (x+1)),$$
$$(\forall t > s_1)(\forall x' < x)(\Gamma^{CU}(x') = \overline{K}(x'))$$

and for every $x' < x$, $\gamma(x')$ does not change after s_1 and

$$(\forall t > s_1)((C \oplus U)[s_1] \upharpoonright (\gamma(x')+1) = (C \oplus U)[t] \upharpoonright (\gamma(x')+1)).$$

By the choice of s_1 and the selection of γ-uses, $\gamma(x)$ at α only moves if there is an $\mathcal{R}$-destroyer strategy $\beta \supset \alpha$ such that $x = w_\beta$, and β or an $\mathcal{R}$-controller strategy γ with $\gamma < \beta$ or $\gamma \supset \beta\hat{\ }0$ acts. If $\beta < f$ then it will not be eligible to act eventually. If $\beta > f$, then w_β goes to infinity. If $\beta \subset f$, then by case assumption, $\beta\hat{\ }1 \subset f$. Let $s_2 > s_1$ be the least stage after which $\beta\hat{\ }1$ is no longer initialized. After stage s_2, any definition of $\Gamma^{CU}(x) = 1$ (if $x \in \overline{K}$) or any correction of $\Gamma^{CU}(x)$ (if $x \in K$) will remain permanent, since neither β nor any such γ will act. Thus $\gamma(x)$ moves only finitely many times.

Notice that we have shown also that, similarly to what happens in the proof of nondensity of the d.c.e. Turing degrees, even in the case when $x \in K$, we define only finitely many axioms $\langle x, F\rangle \in \Gamma$.

CASE 2: $\mathcal{S}$ is satisfied at $f \upharpoonright n$ via some $\mathcal{R}$-destroyer strategy $\beta\hat{\ }0 \subset f$ for all $n > |\beta|$.

We will show that $U = \Delta^C$ where $\Delta = \Delta_\beta$.

Suppose that $U \neq \Delta^C$. Let x be the least counterexample and $s_1 > s_0$ be a stage such that

$$(\forall t > s_1)(U[t] \upharpoonright (x+1) = U[s_1] \upharpoonright (x+1)),$$
$$(\forall x' < x)(\Delta^C(x') = U(x'))$$

and for all $x' < x$, $\delta(x')$ eventually stops moving and

$$(\forall t > s_1)(C[s_1] \upharpoonright (\delta(x')+1) = C[t] \upharpoonright (\delta(x')+1)).$$

Suppose $x \in U$. Since $\beta\hat{}0 \subset f$ there is a (least) stage $s > s_1$ at which we define $\Delta^C(x) = 1$. We argue that the use $\delta(x)$ can only move finitely many times after s. As argued in Case 1, $\delta(x)$ only moves if there are some R-destroyer strategies $\overline{\beta} \supset \beta$ with threshold $u_{\overline{\beta}} \leq x$, and $\overline{\beta}$ or an $\mathcal{R}$-controller strategy γ with $\gamma < \overline{\beta}$ or $\gamma \supset \overline{\beta}\hat{}0$ acts. If $\overline{\beta} < f$ then it only acts finitely often. If $\overline{\beta} > f$ then $u_{\overline{\beta}}$ goes to infinity. If $\overline{\beta} \subset f$, then by case assumption, $\overline{\beta}\hat{}1 \subset f$. Hence $\overline{\beta}$ only acts finitely often. Let $s_2 > s_1$ be a stage after which the longest such $\overline{\beta} \subset f$ is no longer initialized. After s_2, $\Delta^C(x)$ can only be injured if some $z > x$ leaves U with $\delta(z)[s_2] < \delta(x)[s_2]$. As there are only finitely many such z, and when $\delta(z)$ gets redefined (if ever) it will be larger than $\delta(x)$, we only need to redefine $\Delta^C(x)$ finitely many times after s_2.

The argument is similar if $x \notin U$. Following the same notation, eventually, no $\overline{\beta}$ or γ will recover any Δ-axiom $\langle x, F\rangle$. Thus $\Delta^C(x) = 0$.

This completes the proof of the lemma. ∎

LEMMA 5.4: *Every $\mathcal{R}$-requirement is satisfied.*

Proof: By Lemma 4.1, each requirement $\mathcal{R}$ is satisfied at $f \upharpoonright n$ via an $\mathcal{R}$-strategy for almost all n. Let ξ be that $\mathcal{R}$-strategy. By hypothesis, we have $\xi\hat{}1 \subset f$. Let s_0 be minimal such that ξ is not initialized or reset after stage s_0. We distinguish five cases.

CASE 1: ξ is an $\mathcal{R}$-destroyer strategy and stops by itself after stage s_0, say, at stage $s_1 > s_0$. It suffices to show that the computation $\Theta^C[s_1](x) = 1$ is never injured, where $\Theta = \Theta_\xi$ and $x = x_\xi$.

First notice that after stage s_1, no controller γ will want to recover things back to a stage preceding s_1. The reason is that such a γ must be to the left of ξ. If γ has any action at all, it will initialize all nodes to the right, in particular, ξ. Therefore we only need to consider the uses present at stage s_1.

By initialization and our assumption on s_0, only some $\mathcal{S}$-strategy $\eta \subset \xi$ or some $\overline{R}$-destroyer strategy η with $\eta\hat{}0 \subset \xi$ can injure $\Theta^C(x)[s_1]$ by correcting Γ_η or Δ_η on some argument $v \geq w_\xi$ or $v \geq u_\xi$ respectively.

By (6), we only need to consider the nodes which are Σ_3-injured at ξ. Let η be such a node. Suppose that η is Σ_3-injured by the pair ($\mathcal{S}$-strategy) α and ($\mathcal{R}$-destroyer strategy) β. By (4), we have $\min B > |B| > \theta(x)$ at stage s_1 for the $\gamma_\eta(w_\xi)$- or $\delta_\eta(v)$-use block B. Thus η will never injure $\Theta^C(x)[s_1]$.

CASE 2: ξ is an $\mathcal{R}$-destroyer strategy and neither stops by itself, nor is permanently stopped by some fixed $\overline{R}$-controller strategy $\gamma \supseteq \xi\hat{}0$, after stage s_0.

We first claim that ξ is eligible to act at infinitely many stages without being stopped by any $\overline{\mathcal{R}}$-controller strategy $\gamma \supseteq \xi\hat{\ }0$. For the sake of a contradiction, assume there are only finitely many such stages, and let s_1 be the largest such stage. Certainly $s_1 \geq s_0$ by initialization or resetting. Then after stage s_1, no strategy $\eta \supseteq \xi\hat{\ }0$ is eligible to act, and only $\overline{\mathcal{R}}$-controller strategies $\gamma \supseteq \xi\hat{\ }0$ are allowed to act first by ξ or some $\xi' \subset \xi$. Whenever some such γ no longer stops ξ then all strategies $\eta > \gamma$ are initialized, and the next γ' to stop ξ must therefore be $\gamma' < \gamma$ and must have been eligible to act before stage s_1. But there are only finitely many such strategies γ; so either one fixed such γ eventually stops ξ forever or ξ is no longer stopped, a contradiction.

Thus ξ must eventually be stuck waiting for the conditions (1) through (5) in Case 2 of the construction to hold for a fixed $x = x_\xi \in A$. Suppose $\Theta_\xi^C = A$. Thus i_ξ and $\theta_\xi(x + i_\xi)$ are eventually fixed. Since $\xi \subset f$, $\lim s^* = \lim i_\beta = \infty$ for all $\beta \in \mathcal{D}$, and $\lim \min\{|B| : B \in \mathcal{B}_0\} = \infty$. So (1) through (5) hold true at cofinitely many stages, a contradiction.

Case 3: ξ is an $\mathcal{R}$-controller strategy that never takes charge of other strategies after stage s_0.

Suppose $\Theta_\xi^C = A$. Thus $x = x_\xi \in A$ and $\theta_\xi(x)$ is eventually fixed while $\lim i_l = \infty$ for all $l < l_0$, $\lim s^* = \infty$, and $\lim \min\{|B| : B \in \mathcal{B}_2\} = \infty$ since $\xi \subset f$. So (7) through (10) must hold at cofinitely many stages, a contradiction.

Cases 4 and 5: ξ is an $\mathcal{R}$-destroyer strategy that is permanently stopped by some fixed $\overline{\mathcal{R}}$-controller strategy $\gamma \supseteq \xi\hat{\ }0$ after stage s_0; or ξ is an $\mathcal{R}$-controller strategy γ that takes charge of other strategies after stage s_0.

Since $\gamma \leq f$, γ will be initialized or reset after stage s_0 only finitely often, say never after $s_1 \geq s_0$. Then γ takes charge of other strategies forever at some stage $s_* + 1 \geq s_1$. As each U_j for $0 \leq j \leq j_0$ is a 3-c.e. set, γ's parameter l_* can change at most finitely often once γ takes charge of other strategies, so say l_* will not change after (at least) stage $s_2 \geq s_* + 1$.

We show that $x = x_\gamma$ is the witness for diagonalization. By minimality of s_2, γ will ensure (11) and (12) for l_* at stage s_2. By (11), we have

$$\Theta_\xi^C(x)[s_2] = \Theta_\xi^C(x)[s_*] = 1 \neq 0 = A(x).$$

By initialization and our assumption on s_1, only $\mathcal{S}$-strategies $\alpha \subset \xi$ and $\overline{\mathcal{R}}$-destroyer strategies β with $\beta\hat{\ }0 \subseteq \xi$ can possibly destroy the computation $\Theta_\xi^C(x)$ after stage s_2. Moreover, the injury can only happen when correcting Γ_α on an argument $\geq \max\{w_{\beta_l} : l < l_0\}$ or correcting Δ_β on an argument $\geq \max\{u_{\beta_l} : l < l_0\}$ (otherwise, β_{l_0}, and thus γ, would be initialized). Denote the set of these strategies by $\mathcal{D}_0$. We will show that no $\eta \in \mathcal{D}_0$ will destroy $\Theta_\xi^C(x)$ after stage s_2.

Let $\mathcal{D}_1$ be the set of all $\mathcal{S}$-strategies $\alpha \in \mathcal{D}_0$ such that α's requirement is not active at ξ via α, and of all $\mathcal{R}$-destroyer strategies β which are Σ_3-injured at ξ. Consider any $\eta \in \mathcal{D}_1$. Then there is some $\beta_{l'}$ with $l' < l_*$, which kills η's e-operator. Then the $\gamma_\alpha(w_{\beta_{l'}})$- or $\delta_\beta(v)$-use block B (where $v \geq u_{\beta_{l'}}$ is the least such that $\Delta_\beta(v) = 1$ is defined) is in ξ's $\mathcal{B}_0$ (in Case 4), or $\mathcal{B}_2$ (in Case 5), at stage s_*; and by (4) and (7) (in Case 4), or by (9) (in Case 5), we have $\min B > |B| > \theta_\xi(x)[s_*]$. By initialization, no $\overline{R}$-controller strategy $\overline{\gamma} \neq \gamma$ can restore C on B after stage s_*. Now B is in γ's set $\mathcal{B}_3$ (since Γ_α or Δ_β is destroyed by $\beta_{l'}$). Thus by (12), C is permanently changed on B by γ at stage s_2, and thus any $\gamma_\alpha(w)$- or $\delta_\beta(v)$-use block (for $w \geq w_{\beta_{l'}}$ and $v \geq u_{\beta_{l'}}$) that applies after stage s_2 exceeds $\theta_\xi(x)[s_*]$. Thus η cannot destroy $\Theta_\xi^C(x)$ after stage s_2.

Next consider an $\mathcal{S}_j$-strategy α which is active at ξ (for some $j \leq j_0$). The plan is to argue that U_j must have changed below a certain (relatively small) number so that $\gamma(w)$ is lifted beyond $\theta_\xi(x)$. Formally, let l' be minimal such that $\beta_{l'}$ is targeted to destroy Γ_α. As $\mathcal{S}_j$ is active at β_{l_*} we have that $l' > l_*$. Moreover, $a_{j+1} < l_* \leq b_{j+1} = l'$, where a and b are the parameters in the decision algorithm. Thus,

$$U_{j,s} \restriction (y_{l'+1} + 1) \not\supseteq U_{j,s_*} \restriction (y_{l'+1} + 1)$$

for all $s \geq s_2$. We need to show that no correction of $\Gamma_\alpha^{CU_j}(w)[s]$ for $w \geq w_{\beta_{l'}}$ can injure the computation $\Theta_\xi^C(x)[s_2] = \Theta_\xi^C(x)[s_*]$.

At stage $s_* + 1$, $\beta_{l'}$ changes C on the $\gamma_\alpha(w_{\beta_{l'}})$-use block. By the way uses are selected, if $\Gamma_\alpha^{CU_j}(w)$ is defined for the first time at a stage after s_*, then the minimal element in its use block will be fresh at that stage, in particular larger than $\theta_\xi^C(x)[s_*]$. Hence, in this case correction of $\Gamma_\alpha^{CU_j}(w)$ will not be a problem. If $\Gamma_\alpha^{CU_j}(w)$ is first defined at some stage before s^*, then it is permanently destroyed by the C-change at stage s^*. If remains to consider $\Gamma_\alpha^{CU_j}(w)$ which is first defined at some stage between s^* and s_*+1. First observe that by (4) (for $\beta_l, l < l' < l_0$), and by (7) and (9) for γ,

$$y_{l'+1} = \max\{\theta_{\beta_l}(x_\gamma) : l' < l \leq l_0\} < |B| < \min B$$

for the $\gamma_\alpha(w_{\beta_{l'}})$-use block B that $\beta_{l'}$ uses to destroy Γ_α at stage s_*+1. Secondly, by (5) or (10) for $\beta_{l'+1}$

$$(\forall s)(\forall t)(s^* \leq s \leq s_* + 1 \ \&\ t \geq s_2 \Rightarrow U_{j,s} \restriction (y_{l'+1} + 1) \not\subseteq U_{j,t} \restriction (y_{l'+1} + 1)),$$

and no definition using block B ever applies after stage s_2.

Thus no $\mathcal{S}_j$-strategy α which is not Σ_3-injured can destroy $\Theta_\xi^C(x)$ after stage s_2.

Finally, consider an $\overline{R}$-destroyer strategy β_m $(m < l_*)$ which is not Σ_3-injured at ξ. Then some requirement $\mathcal{S}_j$ is satisfied at β_{l_*} via β_m. So by (b) of Lemma 4.3,

$$U_{j,s} \upharpoonright (y_{l_*}+1) \supseteq U_{j,s_*} \upharpoonright (y_{l_*}+1)$$

for all $s \geq s_2$. Thus when γ restores $C \upharpoonright (y_{l_*}+1)$ at stage s_2 (by (11)), we have that for any $v < y_{l_*}+1$, if $\Delta_\beta^C(v)$ is defined by stage s_* then β will not correct it after stage s_2, so it will not change any $C \upharpoonright (\delta(v)+1)$ or, a fortiori, $C \upharpoonright (\theta_\xi(x_\gamma)+1)$. Thus no $\overline{R}$-destroyer strategy β which is not Σ_3-injured at ξ can destroy $\Theta_\xi^C(x_\gamma)$ after stage s_2.

This concludes the proof of Lemma 5.4 and thus Theorem 1.10. ∎

By replacing the function g in the construction by g_n in Lemma 5.2, one can easily obtain a proof for Theorem 1.11.

6. The low n-c.e. e-degrees

We show in this section that no low n-c.e. e-degree can be maximal among the n-c.e. e-degrees, $2 \leq n \leq \omega$. Let us say that an e-operator Ψ is *special* if it satisfies the following property: for every e, j

$$\langle\langle e,j\rangle, F\rangle \in \Psi \Rightarrow F = \emptyset \text{ or } F = \{j\}.$$

(In particular, Ψ is an s-operator, see [13], [4].) Clearly, if Ψ is special and Y is a Δ_2^0 set then Ψ^Y is Δ_2^0. In fact Ψ^Y is n-c.e. if Y is n-c.e.

We have:

THEOREM 6.1: *If $X <_e Y$ are Δ_2^0 sets such that X is low then there exists a special e-operator Ψ such that $X <_e X \oplus \Psi^Y <_e Y$.*

Proof: The proof is a straightforward modification of Gutteridge's proof showing that no nonzero Δ_2^0 e-degree can be minimal, [13] (see also [4]). Let X, Y be given.

THE REQUIREMENTS: The construction aims to build an e-operator Ψ such that the following requirements are satisfied:

$$\mathcal{P}_e : \Psi^Y \neq \Theta_e^X,$$
$$\mathcal{Q}_e : Y \neq \Theta_e^{X\oplus\Psi^Y},$$

where we recall that $\{\Theta_e\}_{e\in\omega}$ is the standard listing of the e-operators, with computable approximations $\{\Theta_{e,s}\}_{e,s\in\omega}$. Let us define a new effective listing $\{\Phi_e\}_{e\in\omega}$ of the e-operators, through suitable computable approximations, as follows. Let

$$\begin{aligned}\Phi_{2e,s} &= \Theta_{e,s},\\ \Phi_{2\langle i,F\rangle+1,s} &= \{\langle x,D\rangle : (\exists D')(\langle x, D\oplus D'\rangle \in \Theta_{i,s}\\ &\qquad\qquad \text{and } D' \subseteq F\cup\omega^{[>i]})\}\end{aligned}$$

where F is (the canonical index of) a finite set and $\omega^{[>i]} = \{\langle j,x\rangle : j > i\}$ (accordingly, $\omega^{[\le i]} = \{\langle j,x\rangle : j\le i\}$). Finally for every e, let $\Phi_e = \bigcup_s \Phi_{e,s}$.

We recall that the e-jump of a set Z is the set $J(Z) = K_Z \oplus \overline{K_Z}$, where $K_Z = \{e : e \in \Theta_e^Z\}$. Since X is low, we have that $J(X) \equiv_e J(\emptyset)$. As clearly $K_\emptyset \equiv_1 \{e : e\in\Phi_e^\emptyset\}$, and thus $J(X)\equiv_e \widehat{K}_\emptyset \oplus \overline{\widehat{K}_\emptyset}$, where $\widehat{K}_\emptyset = \{e : e\in\Phi_e^\emptyset\}$, by the proof of Lemma 4 of [16] there exists a low approximation $\{X_s\}_{s\in\omega}$ to X relatively to the e-operators $\{\Phi_e\}_{e\in\omega}$ and their approximations $\{\Phi_{e,s}\}_{e,s\in\omega}$, i.e., for every e, $\{\Phi_{e,s}^{X_s}\}_{s\in\omega}$ is a Δ_2^0-approximation to Φ_e^X. In particular, for every e,j,

$$(*)\qquad \lim_s \Phi_{e,s}^{X_s}(j) \text{ exists.}$$

This is the Δ_2^0-approximation to X that will be used in this proof. Let $\{Y_s\}_{s\in\omega}$ be any Δ_2^0-approximation to Y.

The construction: The construction is by stages. At stage s we define Ψ_s. Contrary to what we have done in the proofs of Theorem 1.10 and Theorem 1.11, we are more explicit here about mentioning in full detail the stage approximations to the various sets, due to the important role played here by approximations. Let

$$l(e,s) = \min\{x\le s : \Psi_s^{Y_s}(x) \ne \Theta_{e,s}^{X_s}(x)\}$$

and

$$L(e,s) = \min\{x \le s : Y_s(x) \ne \Theta_{e,s}^{X_s\oplus\Psi_s^{Y_s}}(x)\}.$$

By properties of the low approximation to X with which we are working, and by the fact that Ψ is special, we immediately have that $\lim_s l(e,s)$ always exists, either infinite or finite, and

$$\Psi^Y \ne \Theta_e^X \Leftrightarrow \lim_s l(e,s) < \infty$$

as $\Theta_{e,s}^{X_s} = \Phi_{2e,s}^{X_s}$, and thus $\lim_s \Theta_{e,s}^{X_s}(x)$ exists for every x.

We now give the construction:

STEP 0: Let $\Psi_0 = \emptyset$.

STEP $s+1$: For every $e \leq s$,

(a) if $j \leq l(e,s)$ then enumerate $\langle\langle e,j\rangle, \{j\}\rangle$ into Ψ_{s+1};

(b) if $j \leq L(e,s)$ and there exist finite sets E, F, G such that

$$\langle j, E \oplus (F \cup G)\rangle \in \Theta_{e,s}, \quad E \subseteq X_s, \quad F \subseteq \omega^{[\leq e]}, \quad G \subseteq \omega^{[>e]},$$

then choose such a triple of finite sets (in a **consistent** way: this means that at stage s we should choose the least triple such that E has been a subset of X for the longest time. In this way we eventually choose the same triple if there is some triple that eventually shows up at every big enough stage), and enumerate $\langle g, \emptyset\rangle$ into Ψ_{s+1}, for every $g \in G$.

Finally, let Ψ_{s+1} consist of all the elements in Ψ_s plus the elements which have been enumerated into Ψ_{s+1} through (a) or (b). This ends the construction.

Proof that the construction works: Let

$$I_e^P = \{\langle e,j\rangle : \langle e,j\rangle \in \Psi^Y\},$$
$$I_e^Q = \{e : \langle e,\emptyset\rangle \in \Psi\}.$$

We show by induction that for every e

(1) I_e^P and I_e^Q are finite;

(2) $\lim_s l(e,s)$ and $\lim_s L(e,s)$ exist and are finite;

(3) $\mathcal{P}_e$ and $\mathcal{Q}_e$ are satisfied.

Notice that $I_0^Q = \emptyset$. Assume that the claim is true of every $i < e$ and suppose further that I_e^Q is finite. Assume for a contradiction that $\Psi^Y = \Theta_e^X$. Then $l(e,s)$ is unbounded and therefore for every j, $\langle\langle e,j\rangle, \{j\}\rangle \in \Psi$, but for only finitely many j do we have $\langle\langle e,j\rangle, \emptyset\rangle \in \Psi$. Therefore, for all but finitely many j,

$$j \in Y \Leftrightarrow \langle e,j\rangle \in \Psi^Y$$

which implies $Y \leq_e X$, a contradiction. Therefore $\Psi^Y \neq \Theta_e^X$ and $\lim_s l(e,s)$ is finite. Hence the construction enumerates only finitely many numbers of the form $\langle\langle e,j\rangle, \{j\}\rangle$ in Ψ via (a) of the construction. Hence $\mathcal{P}_e$ is satisfied and I_e^P is finite.

By the inductive assumption and what we have just proved, the set $F = \Psi^Y \cap \omega^{[\leq e]}$ is finite.

We now show that $\lim_s L(e,s)$ exists, either infinite or finite, and

$$Y = \Theta_e^{X \oplus \Psi^Y} \Leftrightarrow \lim_s L(e,s) < \infty.$$

To this end suppose that for every $j' < j$ we have that $Y(j') = \Theta_e^{X\oplus\Psi^Y}(j')$ and $\lim_s \Theta_{e,s}^{X_s\oplus\Psi_s^{Y_s}}(j')$ exists. Then there is a stage u such that for every $s \geq u$ we have that $L(e,s) \geq j$ and $\Psi_s^{Y^s}(z) = \Psi_u^{Y^u}(z)$ for every $z \in F$. We first examine the case when $j \in \Theta_e^{X\oplus\Psi^Y}$. In this case $\lim_s \Theta_{e,s}^{X_s\oplus\Psi_s^{Y_s}}(j)$ exists, and since $\lim_s Y_s(j)$ also exists we have that either $\lim_s L(e,s) = j$ if $j \notin Y$, or eventually $L(e,s) > j$ if $j \in Y$. Assume now that $j \notin \Theta_e^{X\oplus\Psi^Y}$. We claim in this case that $j \notin \Phi_{2\langle e,F\rangle+1}^X$. For the sake of a contradiction let us suppose otherwise. Then there exists some triple of finite sets E, F', G such that

$$\langle j, E\oplus(F'\cup G)\rangle \in \Theta_e, \quad E \subseteq X, \quad F' \subseteq F, \quad G \subseteq \omega^{[>e]}.$$

But then eventually the construction selects some such triple making sure that $G \subseteq \Psi^Y$ by enumerating $\langle g, \emptyset\rangle$ into Ψ for every $g \in G$, thus giving $j \in \Theta_e^{X\oplus\Psi^Y}$, a contradiction. Since $j \notin \Phi_{2\langle e,F\rangle+1}^X$, by $(*)$ there exists a stage $u' \geq u$ such that $j \notin \Phi_{2\langle e,F\rangle+1,s}^{X_s}$ for every $s \geq u'$. But then for every such s we have that $j \notin \Theta_{e,s}^{X_s\oplus\Psi_s^{Y_s}}$ since $\Theta_{e,s}^{X_s\oplus\Psi_s^{Y_s}} \subseteq \Phi_{2\langle e,F\rangle+1,s}^{X_s}$. Thus $\lim_s \Theta_{e,s}^{X_s\oplus\Psi_s^{Y_s}}(j)$ exists, and since $\lim_s Y_s(j)$ exists we have that either $\lim_s L(e,s) = j$ if $j \in Y$, or eventually $L(e,s) > j$ if $j \notin Y$.

Next, assume for a contradiction that $Y = \Theta_e^{X\oplus\Psi^Y}$, hence $L(e,s)$ is unbounded. We claim that this implies that

$$Y = \Theta_e^{X\oplus(F\cup\omega^{[>e]})},$$

which would imply that $Y \leq_e X$, a contradiction. Indeed $Y \subseteq \Theta_e^{X\oplus(F\cup\omega^{[>e]})}$, as $\Theta_e^{X\oplus\Psi^Y} \subseteq \Theta_e^{X\oplus(F\cup\omega^{[>e]})}$. On the other hand, if $j \in \Theta_e^{X\oplus(F\cup\omega^{[>e]})}$ then there exists a finite set $G \subseteq \omega^{[>e]}$ such that $j \in \Theta_e^{X\oplus(F\cup G)}$, and the construction makes sure that $G \subseteq \Psi^Y$, thus yielding that $j \in \Theta_e^{X\oplus\Psi^Y}$, i.e., $j \in Y$. Hence $Y \neq \Theta_e^{X\oplus\Psi^Y}$ and $\mathcal{Q}_e$ is satisfied.

Finally, I_{e+1}^Q is finite since $Y \neq \Theta_e^{X\oplus\Psi^Y}$ implies that $\lim_s L(e,s)$ exists and is finite.

This concludes the proof. ∎

We are now ready to conclude:

COROLLARY 6.2: *No low n-c.e. e-degree can be maximal among the n-c.e. e-degrees, for $2 \leq n \leq \omega$.*

Proof: If X is n-c.e., $n \geq 2$, and low, then by the previous theorem there exists a special e-operator Ψ such that $X <_e X \oplus \Psi^{\overline{K}} <_e \overline{K}$. On the other hand $\Psi^{\overline{K}}$ is 2-c.e. Hence $X \oplus \Psi^{\overline{K}}$ is n-c.e. ∎

References

[1] M. M. Arslanov, *Open questions about the n-c.e. degrees,* in *Computability Theory and Its Applications: Current Trends and Open Problems* (M. Lerman P. Cholak, S. Lempp and R. A. Shore, eds.), Contemporary Mathematics **257** (2000), 15–22.

[2] M. M. Arslanov, I. Kalimullin and A. Sorbi, *Density results in the* Δ_2^0 *e-degrees,* Archive for Mathematical Logic **40** (2001), 597–614.

[3] W. C. Calhoun and T. A. Slaman, *The* Π_2^0 *enumeration degrees are not dense,* Journal of Symbolic Logic **61** (1996), 1364–1379.

[4] S. B. Cooper, *Partial degrees and the density problem,* Journal of Symbolic Logic **47** (1982), 854–859.

[5] S. B. Cooper, *Partial degrees and the density problem. II. The enumeration degrees of the* Σ_2 *sets are dense,* Journal of Symbolic Logic **49** (1984), 503–513.

[6] S. B. Cooper, *Enumeration reducibility, nondeterministic computations and relative computability of partial functions,* in *Recursion Theory Week, Oberwolfach 1989* (K. Ambos-Spies, G. Müller and Gerald E. Sacks, eds.), Lecture Notes in Mathematics **1432**, Springer-Verlag, Heidelberg, 1990, pp. 57–110.

[7] S. B. Cooper, L. Harrington, A. H. Lachlan, S. Lempp and R. I. Soare, *The d.r.e. degrees are not dense,* Annals of Pure and Applied Logic **55** (1991), 125–151.

[8] R. L. Epstein, R. Haas and R. L. Kramer, *Hierarchies of sets and degrees below* $\mathbf{0}'$, in *Logic Year 1979–80,* Lecture Notes in Mathematics **859**, Springer-Verlag, Heidelberg, 1981, pp. 32–48.

[9] Yu. L. Ershov, *A hierarchy of sets, I,* Algebra i Logika **7** (1968), 47–74; English Translation: Consultants Bureau, New York, pp. 25–43.

[10] Yu. L. Ershov, *A hierarchy of sets, II,* Algebra i Logika **7** (1968.), 15–47; English Translation, Consultants Bureau, New York, pp. 212–232.

[11] Yu. L. Ershov, *A hierarchy of sets, III,* Algebra i Logika **9(1)** (1970), 34–51; English Translation: Consultants Bureau, New York, pp. 20–31.

[12] R. M. Friedberg and H. Rogers, Jr, *Reducibility and completeness for sets of integers,* Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik **5** (1959), 117–125.

[13] L. Gutteridge, *Some Results on Enumeration Reducibility,* PhD thesis, Simon Fraser University, 1971.

[14] I. Kalimullin, *Splitting properties of n-c. e. enumeration degrees,* Journal of Symbolic Logic **67** (2002), 537–546.

[15] A. H. Lachlan and R. A. Shore, *The n-rea enumeration degrees are dense,* Archive for Mathematical Logic **31** (1992), 277–285.

[16] K. McEvoy and S. B. Cooper, *On minimal pairs of enumeration degrees*, Journal of Symbolic Logic **50** (1985), 983–1001.

[17] J. Myhill, *A note on degrees of partial functions*, Proceedings of the American Mathematical Society **12** (1961), 519–521.

[18] H. Putnam, *Trial and error predicates and the solution to a problem of Mostowski*, Journal of Symbolic Logic **30** (1965), 49–57.

[19] H. Rogers, Jr, *Computing degrees of unsolvability*, Mathematische Annalen **138** (1959), 125–140.

[20] H. Rogers, Jr, *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, New York, 1967.

[21] G. E. Sacks, *A minimal degree below* $\mathbf{0}'$, Bulletin of the American Mathematical Society **67** (1961), 416–419.

[22] G. E. Sacks, *The recursively enumerable degrees are dense*, Annals of Mathematics **80** (1964), 300–312.

[23] C. Spector, *On the degrees of recursive unsolvability*, Annals of Mathematics **64** (1956), 581–592.