The Journal of Symbolic Logic
Volume 85, Number 1, March 2020

ON ISOMORPHISM CLASSES OF COMPUTABLY ENUMERABLE EQUIVALENCE RELATIONS

URI ANDREWS AND SERIKZHAN A. BADAEV

Abstract. We examine how degrees of computably enumerable equivalence relations (ceers) under computable reduction break down into isomorphism classes. Two ceers are isomorphic if there is a computable permutation of ω which reduces one to the other. As a method of focusing on nontrivial differences in isomorphism classes, we give special attention to weakly precomplete ceers. For any degree, we consider the number of isomorphism types contained in the degree and the number of isomorphism types of weakly precomplete ceers contained in the degree. We show that the number of isomorphism types must be 1 or ω, and it is 1 if and only if the ceer is self-full and has no computable classes. On the other hand, we show that the number of isomorphism types of weakly precomplete ceers contained in the degree can be any member of $[0, \omega]$. In fact, for any $n \in [0, \omega]$, there is a degree d and weakly precomplete ceers $E_1, \dots, E_n$ in d so that any ceer R in d is isomorphic to $E_i \oplus D$ for some $i \leq n$ and D a ceer with domain either finite or ω comprised of finitely many computable classes. Thus, up to a trivial equivalence, the degree d splits into exactly n classes.

We conclude by answering some lingering open questions from the literature: Gao and Gerdes [11] define the collection of essentially FC ceers to be those which are reducible to a ceer all of whose classes are finite. They show that the index set of essentially FC ceers is Π^0_3-hard, though the definition is Σ^0_4. We close the gap by showing that the index set is Σ^0_4-complete. They also use index sets to show that there is a ceer all of whose classes are computable, but which is not essentially FC, and they ask for an explicit construction, which we provide.

Andrews and Sorbi [4] examined strong minimal covers of downwards-closed sets of degrees of ceers. We show that if (E_i) is a uniform c.e. sequence of nonuniversal ceers, then $\{\oplus_{i \leq j} E_i \mid j \in \omega\}$ has infinitely many incomparable strong minimal covers, which we use to answer some open questions from [4].

Lastly, we show that there exists an infinite antichain of weakly precomplete ceers.

§1. Introduction. Computable reduction, a natural computability-theoretic analog of borel reduction and first introduced by Ershov [9, 10] as a computable representation for monomorphisms of numbered sets is defined by letting a binary relation R on ω reduce to a binary relation S on ω (written $R \leq_c S$) if there is a computable function f so that for every $x, y \in \omega$, $x \, R \, y$ if and only if $f(x) \, S \, f(y)$. The situation when R and S are equivalence relations, as in the borel theory, is of particular interest. In this article, we continue the trend from [1]–[6] of examining the structure of the set of computably enumerable equivalence relations (ceers) under computable reduction. There has also been a study of the relationship between ceers and the algebraic structures which have the ceer as its domain, see, e.g., [11, 13–16].

Received June 11, 2018.
2010 *Mathematics Subject Classification.* 03D25.
Key words and phrases. computably enumerable equivalence relation, computable reducibility on equivalence relations, halting jump.

0022-4812/20/8501-0004
DOI:10.1017/jsl.2019.39

We also consider isomorphisms defined as follows: $R \cong S$ if there is a computable function f which is a permutation of ω so that xRy if and only if $f(x)Sf(y)$. In this article, we examine the question of how a $\leq_c$-degree splits into isomorphism classes. We show, in particular, that every degree contains either exactly 1 or infinitely many isomorphism classes, but there are degrees with "essentially" any finite number of isomorphism classes. Rigorously, this means that for any n, there are $\leq_c$-classes which contain n ceers $E_0, \dots, E_{n-1}$ which are each nonisomorphic and have no computable classes so that any ceer R in the class is isomorphic to $E_i \oplus D$ where D is a ceer (possibly on a finite domain) comprised of finitely many computable classes.

We also use the idea of a weakly precomplete ceer [6] as the idea of a ceer which is far from having any computable classes. Formally, a ceer E is weakly precomplete if it has no total computable diagonal function, i.e., there is no total computable f so that for every x, $x \not\!\!E\, f(x)$. Every two classes in a weakly precomplete ceer are computably inseparable, so such ceers are far from having computable classes. We examine some further properties of weakly precomplete ceers, but our main use is in constructing the ceers E_i above, which we make weakly precomplete, so that we cannot have $E_i \oplus D_i \cong E_j \oplus D_j$ where D_i and D_j are ceers with computable classes.

In Sections 4, 5, and 6, we also answer some lingering questions in the literature regarding strong minimal covers of some natural subsets of ceers under $\leq_c$, about the set of ceers which are $\leq_c$ a ceer with only finite classes (the essentially FC ceers in [11]), and about antichains of weakly precomplete ceers.

§2. Preliminaries.

We begin with some standard definitions regarding ceers:

DEFINITION 2.1. We let Id_n represent the ceer given by congruence modulo n. Note that any ceer with exactly n equivalence classes is $\equiv_c \mathrm{Id}_n$.

We let Id represent the ceer given by equality, i.e., $x \,\mathrm{Id}\, y$ if and only if $x = y$.

DEFINITION 2.2 (The jump operation). For any ceer R, we define R' to be the ceer given by $x\ R'\ y$ if and only if $x = y$ or $\varphi_x(x)\ R\ \varphi_y(y)$ where both $\varphi_x(x)$ and $\varphi_y(y)$ converge.

The following observation will be helpful for building isomorphisms between ceers.

LEMMA 2.3. *If φ is a reduction of X to Y which is onto the classes of Y, and both X and Y have no finite classes, then $X \cong Y$.*

PROOF. We define a reduction f and a supplementary function g inductively in stages, so $f = \bigcup_i f_i$. We ensure that each f_i is a partial reduction of X to Y, and we ensure $i \in \mathrm{domain}(f_{i+1}) \cap \mathrm{range}(f_{i+1})$. We let $f_0 = \emptyset$. If $i \in \mathrm{domain}(f_i)$, then we let $g_i = f_i$. Otherwise, we enumerate $[\varphi(i)]_Y$ until we see some member j of $[\varphi(i)]_Y \smallsetminus \mathrm{range}(f_i)$. We then add (i, j) to f_i, to form g_i. Now, if $i \in \mathrm{range}(g_i)$, then we let $f_{i+1} = g_i$. Otherwise, we wait until we find some x so that $\varphi(x) \in [i]_Y$ (since φ is onto the classes of Y), and we enumerate $[x]_X$ until we find a member k which is not in $\mathrm{domain}(g_i)$. We then add (k, i) to g_i to form f_{i+1}. On classes, $f = \varphi$, so f is also a reduction of X to Y. By construction, f is a bijection. ⊣

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

Definition 2.4. If E and R are ceers, then $E \oplus R$ is the ceer defined by $x\ E \oplus R\ y$ if and only if either $x = 2n$ and $y = 2m$ and $n\ E\ m$ or $x = 2n + 1$ and $y = 2m + 1$ and $n\ R\ m$.

Observation 2.5. *If E and R are incomparable, then the degree of $E \oplus R$ does not contain a weakly precomplete ceer.*

Proof. Suppose A is weakly precomplete and A reduces to $E \oplus R$ via f, then A reduces to either E or R, since every pair of classes of A are computably inseparable. That is, $\{x \mid f(x) \text{ is odd}\}$ and $\{x \mid f(x) \text{ is even}\}$ provides a separation of two classes, unless one is empty. Thus f reduces A to either E or R. Since $E \oplus R$ does not reduce to E or R, there can be no weakly precomplete A in the same degree as $E \oplus R$. ⊣

Observation 2.6. *The degrees of weakly precomplete ceers are not closed upwards.*

Proof. Let E be weakly precomplete and nonuniversal, which exists by [6]. Let R be any ceer which is $\leq_c$-incomparable with E (see e.g., [3, Theorem 2.1]). Then consider $E \oplus R$. This cannot be a weakly precomplete degree by Observation 2.5. ⊣

We also remind the reader of a useful definition which first appears in [5].

Definition 2.7. A ceer E is self-full if $E <_c E \oplus \mathrm{Id}_1$. Equivalently (see [5]), and motivating the name, E is self-full if whenever φ is a $\leq_c$-reduction of E to itself, φ is onto the classes of E (i.e., for every j, $\mathrm{im}(\varphi) \cap [j]_E$ is nonempty).

We also note the following, which we will use to show hardness of an index set below:

Observation 2.8. *For every binary Π^0_3-predicate $P(x, y)$ there exists a computable binary function g so that $P(x, y) \iff W_{g(x,y)}$ is coinfinite.*

Proof. By [17, Corollary 14-XVI], $\{x : W_x \text{ is cofinite}\}$ is Σ^0_3-complete set. Then, $\{\langle x, y\rangle P(x, y)\} \leqslant_1 \{x : W_x \text{ is coinfinite}\}$ by some computable function f. Define $g(x, y) = f(\langle x, y\rangle)$. ⊣

§3. Isomorphism types inside degrees of ceers. Badaev and Sorbi [6] showed that there are infinitely many isomorphism types of universal weakly precomplete ceers. It is natural to ask whether there are nonuniversal weakly precomplete ceers which are $\leq_c$-equivalent, but not isomorphic. We answer this question, introducing some techniques (especially the strategy for D-requirements) which will appear in the following theorems.

Theorem 3.1. *There are nonisomorphic weakly precomplete ceers which are equivalent and nonuniversal.*

Proof. We construct ceers E, F, and X so that E and F are equivalent, nonisomorphic, weakly precomplete, and $X \not\leq_c E$. During the construction, we will choose sequences of numbers $(a_i)_i \in \omega$ and $(b_i)_{i\in\omega}$, and we satisfy the following requirements:

$$R_{E\to F} : \text{For every pair } i, j,\ i\ E\ j \Leftrightarrow a_i\ F\ a_j.$$
$$R_{F\to E} : \text{For every pair } i, j,\ i\ F\ j \Leftrightarrow b_i\ E\ b_j.$$

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

WP_E^i : If φ_i is a total function, then for some x, $\varphi_i(x)\ E\ x$.

WP_F^i : If φ_i is a total function, then for some x, $\varphi_i(x)\ F\ x$.

D_i : φ_i is not an isomorphism of E to F.

NU_i : φ_i is not a reduction of X to E.

Note that the R-requirements are not subject to injury, but the others can be injured and reinitialized. We begin by describing the strategies for each requirement:

$R_{E\to F}$: We need to do 2 things to satisfy this requirement: choose a_k, and collapse to maintain consistency. For the least k where a_k is not defined, we choose a_k to be a fresh number (i.e., larger than any number mentioned in the construction). We collapse a_i to a_j in F if we see $i\ E\ j$. We will ensure—see Corollary 3.6—that no other requirement collapses a pair a_i and a_j (i.e., $a_i\ F\ a_j$ only happens if we already see $i\ E\ j$).

$R_{F\to E}$: This is symmetric.

WP_E^i : Choose a witness x to be fresh. Wait for $\varphi_i(x)$ to converge. Then E-collapse x with $\varphi_i(x)$. After this collapse, we no longer consider the requirement active.

WP_F^i : This is symmetric.

D_i : Let x, x', and z be fresh. We wait for a stage t where $\varphi_i(x)$ and $\varphi_i(x')$ converge and $\varphi_i(y) = z$ for some y. If $\varphi_i(x)\ F^t\ \varphi_i(x')$, then we do nothing further (we will see that, unless injured, $x \not\!\!E\ x'$) and no longer consider the requirement active. Otherwise, by possibly reversing x and x', we may assume that $\varphi_i(x) \not\!\!F^t\ z$. Let $w = \varphi_i(x)$. We collapse $x\ E\ y$. We will see below that (unless injured) $z \not\!\!F\ w$. After this collapse, we no longer consider the requirement active.

NU_i : Take two fresh numbers x and y. Wait for $\varphi_i(x)$ and $\varphi_i(y)$ to converge. If $\varphi_i(x) \not\!\!E\ \varphi_i(y)$, then collapse $x\ X\ y$. We will see below that reinitializing lower priority requirements will suffice to guarantee that $\varphi_i(x) \not\!\!E\ \varphi_i(y)$ remains true.

Construction. We fix some priority ordering in order type ω of the WP-, D-, and NU-requirements. As they are not subject to injury, we do not include $R_{E\to F}$- and $R_{F\to E}$-requirements in our priority order. We deal with each of D-, WP-, and NU-requirements at infinitely many stages, one at every stage $s > 0$ of the construction. And we deal with the R-requirements at each stage of the construction. WP_E^i- and WP_F^i-requirements can have a parameter x. D_i-requirements can have parameters x, x', and z. NU_i-requirements can have parameters x and y. When a requirement is initialized, each parameter is set to be undefined and the requirement is set to be active. We say that a requirement requires attention if any of its parameters are undefined, or if it is an active WP^i-requirement and $\varphi_i(x)$ has converged, or it is an active D_i-requirement and $\varphi_i(x), \varphi_i(x')$ converge and some $\varphi_i(y) = z$, or if it is an active NU_i-requirement and $\varphi_i(x)$ and $\varphi_i(y)$ have both converged.

When a strategy for a WP-, D-, or NU-requirement acts, it reinitializes all lower priority D-, WP-, or NU-requirements. Any reinitialized requirement becomes active immediately.

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

Stage 0. Initialize each WP-, D-, and NU-requirements. Set $E^0 = \mathrm{Id}$, $F^0 = \mathrm{Id}$.

Stage $s+1$. Let $s = \langle s_0, s_1\rangle$ and let Q be the strategy with priority equal to s_0. If Q does not require attention, then end the stage. Otherwise, we say that Q acts, thus reinitializing all lower priority requirements. We consider cases depending on which type of requirement Q is:

Case 1. Suppose Q is a WP^i_E-requirement. If the parameter x is not yet defined, define it to be a fresh number (in particular $[x]_{E^s} = \{x\}$ and x is not a parameter of any other requirement). Otherwise, we have x defined and $\varphi_i(x)$ converged. Then we E-collapse x with $\varphi_i(x)$ and declare WP^i_E to be inactive.

Case 2. Suppose Q is a WP^i_F-requirement. We act exactly as in case 1, but with F-collapse.

Case 3. Suppose Q is a D_i-requirement. If x, x', and z are undefined, then select them to be distinct fresh numbers. Otherwise, we have $\varphi_i(x), \varphi_i(x')$ converged and $\varphi_i(y) = z$ for some y. If $\varphi_i(x) F \varphi_i(x')$, we declare D_i to be inactive and do nothing else. Otherwise, by possibly switching the roles of x and x', we may assume that $\varphi_i(x) \not{F^s}\, z$. We E-collapse x with y and declare the requirement D_i to be inactive.

Case 4. Suppose Q is a NU_i-requirement. If x and y are not defined, then we select them to be fresh numbers. Otherwise, we have $\varphi_i(x)$ and $\varphi_i(y)$ both converged. If $\varphi_i(x)\ E\ \varphi_i(y)$, then we declare NU_i to be inactive. Otherwise, we X-collapse x and y and declare NU_i to be inactive.

In any case, we finish the stage with the following:

Coding Step. Define a_s and b_s to be fresh numbers. Lastly, if we have E-collapsed x and y, and a_x and a_y are defined, then F-collapse a_x and a_y. Similarly, if we have F-collapsed x and y, and b_x and b_y are defined, then E-collapse b_x and b_y. We apply this as many times as necessary, but at stage $s+1$, we have only defined finitely many values of a_x and b_x, so this only causes finitely many collapses.

Verification. We proceed through a sequence of lemmas to show that all requirements are satisfied.

Definition 3.2. Let Q be a WP-, D-, or NU-requirement. We say that x is a Q-number at stage s if x is a parameter of an active Q-requirement.

We say that x is an $R_{E\to F}$-number at stage s if it is defined to be a_i for some i so that i is the least member of $[i]_E$ at stage s.

We say that x is an $R_{F\to E}$-number at stage s if it is defined to be b_i for some i so that i is the least member of $[i]_F$ at stage s.

In each case, we say that the number x is active at stage s.

Lemma 3.3. *Let x and y be distinct active numbers at stage s. Then $x \not{E^s}\, y$ and $x \not{F^s}\, y$.*

Proof. This is clearly true at stage 0. Suppose $s+1$ is the least stage at which this lemma fails. Let x, Q_1, y, and Q_2 witness this. Let us consider the action at stage s which brought about this situation. At stage s, we must have done more than just defining new parameters because all new parameters are chosen to be fresh. In particular, if z is fresh, then $[z]_{E^s} = [z]_{F^s} = \{z\}$, so it cannot contribute to violating our lemma.

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

There are two parts of the construction at stage s: The action in each of the 4 cases, and then the coding step. We verify that after each of these actions, we have not violated our lemma.

In each of the 4 cases where we can cause a collapse, we have a requirement Q-which collapses one of its parameters z to some other element w. We then declare Q to be inactive. By inductive hypothesis, $[z]_E$ and $[z]_F$ contain only one active number, namely, z. Thus, since z is not active at stage $s+1$ since Q becomes inactive, we have added no new active numbers to $[w]_E$ or $[w]_F$.

Lastly, we have to check that our collapses during the coding step do not cause us to violate this lemma. These are of the form of F-collapsing a_i with a_j if we have E-collapsed i with j or of the form of E-collapsing b_i with b_j if we have F-collapsed i with j. We consider the former case as the latter case is the same. We can assume that prior to E-collapsing i with j, both were least in their E-classes. Thus, both a_i and a_j were active. It follows that $[a_i]_F$ and $[a_j]_F$ had only one active element, namely, a_i and a_j. But since i and j have collapsed, one has stopped being active. So the newly formed class $[a_i]_F \cup [a_j]_F$ still contains only one active element. ⊣

Lemma 3.4. *Let $x = b_i$. Then for every $s > i$ there exists a $j \leq i$ so that $xE^s b_j$ and b_j is active at stage s.*

Let $x = a_i$. Then for every $s > i$ there exists a $j \leq i$ so that $xF^s a_j$ and b_j is active at stage s.

Proof. We prove the first claim. Let j be the least number in $[i]_{F^s}$. Then $j \leq i$ and b_j is active. By the coding step of our construction, $i\ F^s\ j$ implies $b_i\ E^s\ b_j$. ⊣

Lemma 3.5. *Let x be a number mentioned before stage s. Suppose that x is not E^s-equivalent to any active number at stage s. Then at all stages $t > s$, x is not E^t-equivalent to any active number at stage t.*

Similarly for F-equivalence.

Proof. Suppose otherwise, and consider the first stage $t > s$ at which x becomes E^t-equivalent to an active number at stage t. This cannot be caused by an assignment of parameters, since all parameters are assigned to be fresh. By the same analysis as in Lemma 3.3, any active z which is collapsed with x must simultaneously become inactive. Similarly, this cannot be caused by collapsing b_i with b_j for the sake of coding because x cannot already be in $[b_i]_E$ or $[b_j]_E$, since these each contain active members. ⊣

Corollary 3.6. *For every $i, j < s$, $i\ E^s\ j$ if and only if $a_i\ F^s\ a_j$. For every $k, l < s$, $k\ F^s\ l$ if and only if $b_k\ E^s\ b_l$.*

Proof. We prove only the first claim as the second is symmetric. By the coding step, $i\ E^s\ j$ implies that $a_i\ F^s\ a_j$. To see the reverse, suppose that $i \not{E^s} j$ and let i_0 be least in $[i]_{E^s}$ and j_0 be least in $[j]_{E^s}$. It follows by the coding step that $a_{i_0}\ F^s\ a_i$ and $a_{j_0}\ F^s\ a_j$. Then a_{i_0} and a_{j_0} are both active numbers at stage s. It follows by Lemma 3.3 that they cannot be F^s-equivalent. Thus $a_i \not{F^s} a_j$. ⊣

It follows that the R-requirements are satisfied.

Lemma 3.7. *Every requirement is reinitialized only finitely often.*

Proof. Straightforward by induction in priority of the requirements. ⊣

Lemma 3.8. *Suppose that x and y are numbers considered before stage s and $x \not{E^s} y$. Suppose that Q is a requirement which is deactivated at stage s (thus all lower*

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

priority requirements are reinitialized at stage s). Suppose further that no requirement of higher priority than Q acts after stage s. Then $x \not\!\!E\, y$. Similarly for F.

Proof.

Case 1. Neither x nor y is E^s-equivalent to any active number. Then by Lemma 3.5, this is true at every $t > s$. Then at any stage $t > s$ we cannot collapse x with y because neither are equivalent to any active numbers.

Case 2. Suppose that either x or y is E^s-equivalent to a $\hat{Q}$-number at stage s for $\hat{Q}$ a WP-, D-, or NU-requirement of higher priority than Q. WLOG, we suppose this is true of x. Suppose towards a contradiction that $x\,E\,y$. Let $t > s$ be the stage at which we cause this collapse. Since $\hat{Q}$ does not act after stage s, we know that x is also E^t-equivalent to a $\hat{Q}$-number at stage t, and thus cannot be E^t-equivalent to any other active number by Lemma 3.3. Thus, the collapse must be caused by an active number E^t-equivalent to y.

Case 2a. Suppose that y is not E^s-equivalent to an active number at stage s. Then by Lemma 3.5, this is true at stage t also, so the collapse cannot occur at stage t.

Case 2b. Suppose that y is E^s-equivalent to b_j for some j. Then by Lemma 3.4, y is E^t-equivalent to some active b_k for some k. Thus, it is not equivalent to a Q'-number at stage t for Q' any WP-, D-, or NU-requirement. Since x cannot be equivalent to any b_l (since Lemma 3.4 shows that it would then be E^t-equivalent to two active numbers contradicting Lemma 3.3), the collapse can also not occur due to the coding.

Case 2c. Suppose that y is also E^s-equivalent to a Q'-number at stage s for Q' a WP-, D-, or NU-requirement of higher priority than Q. Then, y is also E^t-equivalent to a Q'-number at stage t, and thus cannot be E^t-equivalent to any other active number by Lemma 3.3. Thus, since neither Q' nor $\hat{Q}$ can act at stage t, we cannot cause the collapse at stage t.

Case 3. Suppose that $x\ E^s\ b_i$ and y is not E^s-equivalent to any active number. Then Lemma 3.4 shows that $[x]_{E^t}$ contains an active number for every $t > s$ whereas Lemma 3.5 shows that y is never E^t-equivalent to an active number, so x and y can not be E^t-equivalent for any $t > s$.

Case 4. We have $x\ E^s\ b_i$ and $y\ E^s\ b_j$ for some i, j. Then the only cause of the collapse of $x\,E\,y$ is due to the coding step, since neither can ever be E^t-equivalent to any other active number. But then we can consider why we collapse $i\ F\ j$. By Case 3, the only possibility is that this, in turn was due to Case 4, namely, due to a coding step. But the coding step at any given stage is finite and originates in a collapse for a WP-, D-, or NU-requirement, which we have ruled out in the cases above. ⊣

Lemma 3.9. *WP^i_E- and WP^i_F-requirements are satisfied.*

Proof. Consider the last time the requirement is reinitialized. When it next chooses its witness x, this choice is permanent. If $\varphi_i(x)$ does not converge, then the requirement is satisfied. Otherwise, once it converges, the requirement will act (since no higher priority requirement can act) by collapsing x with $\varphi_i(x)$ satisfying the requirement. ⊣

Lemma 3.10. *The D_i-requirement is satisfied.*

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

PROOF. Consider the last time the requirement is reinitialized. When it next chooses its witnesses x, x', *and* z, this choice is permanent. If $\varphi_i(x)$ or $\varphi_i(x')$ does not converge or no $\varphi_i(y)$ converges to equal z, then φ_i is not a bijection and the requirement is satisfied. Now, suppose $\varphi_i(x)$ and $\varphi_i(x')$ converge and $\varphi_i(y)$ converges to equal z, and let t be the first stage after this when this requirement is next considered by the construction. If $\varphi_i(x) F^t \varphi_i(x')$, the requirement does nothing. As x and x' were active at stage t, they were not E^t-equivalent. By Lemma 3.8, we see $x \not{E} x'$. Otherwise, (possibly after reversing x and x'), we have $\varphi_i(x) = w \not{F^s} z$. Then we E-collapsed x with y. It suffices to show that $w \not{F} z$. Since z is not equivalent to any a_i, w and z do not collapse at stage t. Since w and z are certainly considered at stage t, the D_i-requirement becomes inactive at stage t, and we have supposed that no higher priority requirement acts after stage t (as this would reinitialize the D_i-requirement), Lemma 3.8, guarantees that $w \not{F} z$. ⊣

LEMMA 3.11. *The requirement NU_i is satisfied.*

PROOF. Consider the last time that NU_i is reinitialized. When it next picks its witnesses $x andy$, this is permanent. If $\varphi_i(x)$ or $\varphi_i(y)$ never converge, then the requirement is satisfied. Otherwise, consider the next stage s where NU_i acts. If $\varphi_i(x) E^s \varphi_i(y)$ then simply not X-collapsing x and y guarantees that the requirement is satisfied. So, we must consider the case that $\varphi_i(x) \not{E^s} \varphi_i(y)$ and we must show that $\varphi_i(x) \not{E} \varphi_i(y)$. This follows immediately by Lemma 3.8 ⊣

Thus, every requirement eventually succeeds, and we have built ceers E and F as needed. ⊣

THEOREM 3.12. *There are nonuniversal weakly precomplete ceers E_i for $i \in \omega$ so that they are equivalent and pairwise nonisomorphic.*

PROOF. This is the same argument as the previous theorem with no new complications. We construct infinitely many ceers, but the requirements each mention at most two ceers and are handled with strategies identical to the previous argument. ⊣

DEFINITION 3.13. For a ceer E, we define $N(E)$ to be the number of isomorphism types inside the $\leq_c$-degree of E. We define $N^*(E)$ to be the number of isomorphism types of weakly precomplete ceers inside the $\leq_c$-degree of E.

THEOREM 3.14. *$N(E) = 1$ if and only if E is self-full and has no computable classes. Otherwise, $N(E) = \omega$.*

PROOF. We prove the theorem in cases:

LEMMA 3.15. *If E is self-full and has no computable classes then $N(E) = 1$.*

PROOF. Suppose $X \equiv_c E$. Then consider the reductions $E \leq_c X \leq_c E$. Since E is self-full, it follows that the composed reduction is onto the classes of E, therefore, the reduction $E \leq_c X$ is onto the classes of X. Thus, X cannot have any computable classes either, so all of the classes of X are infinite. Thus $E \cong X$ by Lemma 2.3. ⊣

LEMMA 3.16. *If E has no computable classes and is non–self-full, then $N(E) = \omega$.*

PROOF. Consider the set of equivalence relations $E \oplus \mathrm{Id}_n$ for various n. These are all nonisomorphic because $E \oplus \mathrm{Id}_n$ has exactly n computable classes, yet they are all $\leq_c$-equivalent by non–self-fullness of E. ⊣

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

LEMMA 3.17. *If E has a computable class, then $N(E) = \omega$.*

PROOF. Suppose, towards a contradiction, that $N(E) = k$ and let $E_0, E_1, \ldots, E_{k-1}$ be representatives of every isomorphism type in the degree of E.

CLAIM 3.18. *For some $i < k$, there are infinitely many m so that E_i has a finite class of size m.*

PROOF. For each $m \in \omega$, let R_m be the ceer formed by replacing a computable class in E by a class of size m. Let $F(m)$ be the $i < k$ so that $R_m \cong E_i$. By the pigeonhole principle, there is an infinite set of m on which F is constant. ⊣

Without loss of generality, we assume E_0 has this property.

CLAIM 3.19. *For all positive $n \in \omega$, there is an E_i with infinitely many finite classes, but none of size $\leq n$.*

PROOF. For each $n > 0$, consider the ceer Q_n formed by taking E_0 and fattening every point by n numbers. That is, we say $x \; Q_n \; y$ if and only if $\lfloor \frac{x}{n} \rfloor \; E_0 \; \lfloor \frac{y}{n} \rfloor$. ⊣

For each $i < k$, let $l_i \in \omega$ be greater than the size of the smallest finite class of E_i, if there is one, and 1 otherwise. Applying the previous claim with $n = \Sigma_{i<k} l_i$ gives Q_n, a new isomorphism type in the degree of E. ⊣

It may seem like a very weak argument that $N(E)$ is infinite in Lemmas 3.16 and 3.17. One might argue that appending finitely many computable classes to E may yield a new isomorphism type, but it does not give a substantively different isomorphism type. We will see that for some ceers E, the only way to produce other isomorphism types in the same degree is to append finitely many computable classes to E.

COROLLARY 3.20. *There is a weakly precomplete ceer E so that for any X, $X \equiv_c E$ implies $X \cong E$.*

PROOF. It is shown in [6] (or see Theorem 6.3) that there are weakly precomplete ceers E which are dark, i.e., $\mathrm{Id} \not\leq_c E$. It is shown in [5] that all dark ceers are self-full. Lastly, since the classes of weakly precomplete ceers are computably inseparable, no class can be computable. So, a dark weakly precomplete ceer E has $N(E) = 1$ by Lemma 3.15. ⊣

THEOREM 3.21. *The range of N^* is $[0, \omega]$.*

PROOF. The fact that 0, 1, and ω are in the range follows from Observation 2.5, Lemma 3.15, and Theorem 3.12. For $n \in (1, \omega)$, we give the following construction:

We build ceers $E_0, \ldots, E_n$, and we will make $E_0 = E_n$. Towards this, whenever we give instructions to E_0-collapse some pair, it is understood that we simultaneously E_n-collapse and vice versa. We also build functions π_i for $i = 1, \ldots, n$ so that π_i reduces E_{i-1} to E_i. Thus $\pi = \pi_n \circ \pi_{n-1} \circ \cdots \circ \pi_1$ is a reduction of E_0 to itself. We attempt to ensure that the set of weakly precomplete ceers in the degree of E_0 is exactly $\{E_0, \ldots, E_{n-1}\}$, and these are pairwise nonisomorphic, thus $N^*(E_0) = n$. We build these ceers with the following requirements (where we consider $i, i' < n$ distinct and any $j, k \in \omega$):

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

C^i : Pick numbers a_j^{i+1} for each j so that $j\ E_i\ k$ if and only if $a_j^{i+1}\ E_{i+1}\ a_k^{i+1}$.

WP_j^i : If φ_j is a total function, then it has an E_i-fixed point.

$D_j^{i,i'}$: φ_j is not an isomorphism from E_i to $E_{i'}$.

$T_{j,k}^i$: If W_j intersects infinitely many E_i-classes which do not contain any element of the form a_l^i, then W_j intersects $[k]_{E_i}$.

S_j : If φ_j is a reduction of E_0 to E_0, then for some k, $[\mathrm{im}(\varphi_j)]_{E_0} \cap \Big([\mathrm{im}(\pi^k)]_{E_0} \smallsetminus [\mathrm{im}(\pi^{k+1})]_{E_0}\Big)$ contains infinitely many classes.

See Lemmas 3.32 and 3.33 for why the requirements, especially $T_{j,k}^i$ and S_j, suffice for the theorem. Intuitively, π gives us a nested layering of E_0 by smaller copies of itself, and S_j shows that any ceer R equivalent to E_0 must in its reduction to E_0 intersect one annulus infinitely. We then stratify that annulus in terms of the E_i's and see that one of these strata must be hit infinitely. Then $T_{j,k}^i$ shows that the entirety of this copy of E_i must be hit, which is enough for us to analyze the ceer R.

We enumerate all WP, D, T and S-requirements in order type ω as $Q_0 \prec Q_1 \cdots$.

At this point, the strategies for C^i, WP_j^i, and $D_j^{i,i'}$ should be familiar. We highlight the strategies for $T_{j,k}^i$ and S_j and their conflict.

$T_{j,k}^i$*- strategy*. Wait for W_j to enumerate a number x which is not E_i-equivalent to any number mentioned by a higher priority requirement and is also not E_i-equivalent to any element of the form a_l^i. Then collapse $k\ E_i\ x$.

S_j*-strategy*. Througout the description, we let l be maximal so that y^l is defined.

Step 1. Pick new y^0, y^1, and keep $y^0 \not\mathrel{E_0} y^1$ and that neither y^0 nor y^1 will ever be equivalent to an element of the form a_k^n for any k (this will be automatic via Lemma 3.25, which is analogous to Lemma 3.3). Wait for $\varphi_j(y^0)$ and $\varphi_j(y^1)$ to converge. Once this happens, go to Step 2.

Step 2. If for some k we have $\{[\varphi_j(y^0)]_{E_0}, [\varphi_j(y^l)]_{E_0}\} = \{[\pi^k y^0]_{E_0}, [\pi^k y^l]_{E_0}\}$, then go to Step 3. Otherwise, we have two cases: If $\varphi_j(y^0)\ E_0\ \varphi_j(y^l)$ already, then we simply maintain that $y^0 \not\mathrel{E_0} y^l$ and do nothing. Otherwise, we collapse y^0 with y^l, and we will not be forced to collapse $\varphi_j(y^0)$ with $\varphi_j(y^l)$. We reinitialize lower priority requirements to ensure that they will not cause $\varphi_j(y^0)$ and $\varphi_j(y^l)$ to E_0-collapse. We do nothing further.

Step 3. Choose a new number y that is not equivalent to any element of the form a_k^n and assign this to be y^{l+1}. Wait for $\varphi_j(y^{l+1})$ to converge, then go back to Step 2 (with the newly increased value of l).

The possible outcomes of one S_j strategy are either infinite cycling through Steps 2 and 3, or it gets stuck in Step 1, 2, or 3. If it cycles through Steps 2 and 3 infinitely often, this will force that for a single k, every m satisfies $\varphi_j(y^m)\,E_0\,\pi^k y^m$ (see Lemma 3.30). Furthermore, since $y^m \not\mathrel{E_0} \pi(d)$ for every d, we have that $\pi^k y^m \not\mathrel{E_0} \pi^{k+1} d$ for every d. Thus this gives that $[\mathrm{im}(\varphi_j)]_{E_0} \cap [\mathrm{im}(\pi^k)]_{E_0} \smallsetminus [\mathrm{im}(\pi^{k+1})]_{E_0}$ is infinite as needed. If it gets stuck in Step 1 or Step 3, then φ_j is not total, and if it gets stuck in Step 2, then we diagonalize ensuring that φ_j is not a reduction of E_0 to itself.

Now, note that S_j has infinitely many parameters if it cycles through Steps 2 and 3 infinitely often, which is inconsistent with $T_{j,k}^i$-strategies (it causes no problems to D or WP-strategies as these can only cause a collapse involving at least one new number, whereas $T^i_{j,k}$-strategies use numbers chosen by W_j). In particular, if W_j enumerates $\{y^m \mid m \in \omega\}$ chosen by a higher priority S_j-requirement, it would not be able to be satisfied. This problem is fixed by allowing a high enough priority $T^i_{j,k}$-requirement, say it is requirement Q_p to only respect values y^m for $m < p$. In other words, the lower priority T-requirement may steal the S-requirement's parameter y^k for $k \geq p$. Still, only finitely many T-strategies may steal this parameter, so if S cycles through 2 and 3 infinitely often, it will eventually define each y^m.

The remainder of the proof is handled via the usual priority machinery.

Construction. We build equivalence relations $E^s_0, E^s_1, \dots, E^s_{n-1}$ as approximations to the equivalence relations $E_0, E_1, \dots, E_{n-1}$. WP-requirements can have a parameter x. D-requirements can have parameters x, x', and z. T-requirements can have a parameter r. S-requirements can have, as parameters, a finite sequence of numbers $y^0, \dots, y^l$.

We say that a WP-requirement demands attention if it is active and either x is undefined or $\varphi_j(x)$ has converged. We say that a D-requirement requires attention if it is active and either its parameters are undefined or $\varphi_j(x)$, $\varphi_j(x')$ have converged and some $\varphi_j(y) = z$. We say a $T^i_{j,k}$-requirement requires attention if it is active and either r is undefined or some number x has been enumerated into W_j so that x is not E^s_i-equivalent to any a^i_l or any number less than r. We say that an S_j-requirement requires attention if it is active and either y^0 is undefined or, for l being greatest so that y^l is defined, we have both $\varphi_j(y^0)$ and $\varphi_j(y^l)$ converged and either $\{[\varphi_j(y^0)]_{E_0}, [\varphi_j(y^l)]_{E_0}\} = \{[\pi^k y^0]_{E_0}, [\pi^k y^l]_{E_0}\}$ for some k or $\varphi_j(y^0) \not{E_0}\, \varphi_j(y^l)$.

When a requirement is initialized, each parameter is set to be undefined and the requirement is set to be active. For a WP-, D-, T-, or S-requirement Q, let $\#(Q)$ be the number m so that the requirement is Q_m. At stage $s + 1 = \langle m, k\rangle$, we consider the requirement Q_m; thus, each requirement is considered infinitely often.

Stage 0. Initialize all requirements. For every $i < n$, set $E^0_i = \mathrm{Id}$.

Stage $s + 1$. Denote by $\mathcal{Q}$ the requirement that we consider at stage $s + 1$. If $\mathcal{Q}$ does not require attention, then go to the next stage, otherwise we execute the strategy for $\mathcal{Q}$ below, followed by the coding step:

Case 1. $\mathcal{Q}$ is the WP^i_j-requirement for some $i < n$ and $j \in \omega$.

Case 1.1. If a parameter x for WP^i_j is not defined, then choose x to be a fresh number. Reinitialize all lower priority requirements.

Case 1.2. If x is the parameter for WP^i_j and $\varphi^s_j(x)$ has converged, then collapse $x E^{s+1}_i \varphi_j(x)$, reinitialize all lower priority requirements, and declare the WP^i_j-requirement to be inactive.

Case 2. $\mathcal{Q}$ is the $D^{i,i'}_j$-requirement with $i, i' < n$, $i \neq i'$, and $j \in \omega$.

Case 2.1. If no parameters for $D^{i,i'}_j$ are defined, then pick three fresh numbers x, x', and z and set them to be the parameters for $D^{i,i'}_j$. Reinitialize all lower priority requirements.

Case 2.2. If the parameters x, x', and z for $D^{i,i'}_j$ are defined and $\varphi^s_j(x)$ *and* $\varphi^s_j(x')$ are converged, and for some y, $\varphi^s_j(y) = z$, then: If $\varphi_j(x)$ $E^s_{i'}$

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

$\varphi_j(x')$, we declare D_i to be inactive and do nothing else. Otherwise, by possibly switching the roles of x and x', we may assume that $\varphi_i(x) \not\mathrel{E^s_{i'}} z$. We E_i-collapse x with y and declare the requirement $D_j^{i,i'}$ to be inactive. We reinitialize all lower priority requirements.

CASE 3. $\mathcal{Q}$ is the $T^i_{j,k}$-requirement with $i < n$ and $j, k \in \omega$.

CASE 3.1. If $k \in [W_e^s]_{E_{i,s}}$ then declare $T^i_{e,k}$-requirement to be inactive.

CASE 3.2. If $k \notin [W_e^s]_{E_{i,s}}$ and r is not defined, then choose r to be a fresh number. Reinitialize all lower priority requirements.

CASE 3.3. If there is an x in W_j so that x is not E_i-equivalent to any number less than r and also not E_i-equivalent to a^i_l for any l, then E_i-collapse x with k. We declare $\mathcal{Q}$ to be inactive and reinitialize all lower priority requirements. If this number x is already E_i-equivalent to a higher priority S_j-requirement's parameter y^m, then we undefine the S_j-requirement's parameters $y^{m'}$ for every $m' \geq m$. (In Lemma 3.22, we will see that this can only happen if $m > \#(Q)$.)

CASE 4. $\mathcal{Q}$ is the S_j-requirement. If S_j does not require attention, then do nothing. Otherwise:

CASE 4.1. The parameter y^0 is not defined. Then choose y^0 and y^1 to be fresh numbers. Reinitialize all lower priority requirements.

If the parameter y^0 is defined, then let l be largest so that the parameter y^l is defined.

CASE 4.2. We have $\varphi_j(y^0)$ and $\varphi_j(y^l)$ both converged and $\{[\varphi_j(y^0)]_{E_0^s}, [\varphi_j(y^l)]_{E_0^s}\} = \{[\pi^k y^0]_{E_0^s}, [\pi^k y^l]_{E_0^s}\}$ for some k. In this case, define the parameter y^{l+1} to be fresh and initialize all lower priority requirements $\mathcal{R}$ so that $\#(\mathcal{R}) \geq l$.

CASE 4.3. We have $\varphi_j(y^0)$ and $\varphi_j(y^l)$ both converged and $\{[\varphi_j(y^0)]_{E_0^s}, [\varphi_j(y^l)]_{E_0^s}\} \neq \{[\pi^k y^0]_{E_0^s}, [\pi^k y^l]_{E_0^s}\}$ for any k, and $\varphi_j(y^0) \not\mathrel{E_0^s} \varphi_j(y^l)$. Then E_0-collapse y^0 and y^l and reinitialize all lower priority requirements. Declare S_j to be inactive.

CODING STEP. For each $i = 1, \dots n$, choose distinct fresh numbers to be a^i_s. For each $i < n$, if we have E_i-collapsed j with k, and a^{i+1}_j and a^{i+1}_k are defined, then E_{i+1}-collapse a^{i+1}_j and a^{i+1}_k. Note that since we have only finitely many a^{i+1}_j-values defined, this causes only finitely many collapses.

VERIFICATION.

LEMMA 3.22. *If a T-requirement undefines an S-requirement's parameter y^m, then* $\#(T) < m$.

PROOF. It must be that $x \mathrel{E_i} y^m$, but $x \not\mathrel{E_i} z$ for every $z \leq r$. Thus $y^m > r$. In particular, y^m was chosen to be a parameter after the T-requirement assigned the parameter r. So when y^m was assigned in Case 4.2, the T-requirement was not reinitialized, thus $\#(T) < m$. ⊣

LEMMA 3.23. *Each requirement is reinitialized only finitely often.*

PROOF. Suppose towards a contradiction that Q is the highest priority requirement reinitialized infinitely often. There must be a single higher priority requirement $\mathcal{R}$ which reinitializes Q infinitely often. Let s be a stage after which $\mathcal{R}$ is never reinitialized. If $\mathcal{R}$ is a WP-, D-, or T-requirement, then it can act only once more, after which it becomes inactive. Thus $\mathcal{R}$ must be an S-requirement. For $\mathcal{R}$ to injure Q, it does so in Case 4.1, 4.2, or 4.3 infinitely often. It can do so in Cases 4.1 and 4.3 only once after stage s, as it is not reinitialized after stage s. Thus, it must infinitely often injure Q via Case 4.2. In this case, it infinitely often increases the maximal l for which a^l is defined. We need only show that after finitely many stages, this l will permanently exceed $\#(Q)$. Otherwise, infinitely often, we must have a T-requirement with $\#(T) < \#(Q)$ which undefines $\mathcal{R}$'s parameter $y^{\#(Q)}$. But each such $\#(T)$ can do this only once. This is because a $T^i_{j,k}$-requirement acts in Case 3.3 by making $k \in [W^s_e]_{E_{i,s}}$. But then it can never be in Case 3.3 again, as it would be in Case 3.1 instead. Thus, each T-requirement with $\#(T) < \#(Q)$ can undefine $\mathcal{R}$'s parameter $y^{\#(Q)}$ at most once, so it will eventually be permanently defined and $\mathcal{R}$ will not reinitialize Q after that. ⊣

Definition 3.24. Let Q be a WP^i_j-requirement. We say that x is an i-active Q-number at stage s if x is a parameter of Q at stage s. Let Q be an active $D^{i,i'}_j$-requirement with parameters $x, x', and\ z$. Then we say that x, x' are i-active Q-numbers at stage s and z is an i'-active Q-number at stage s. Let Q be an S-requirement and x be a parameter of Q. We say that x is a 0-active and n-active Q-number at stage s.

We say that x is an $i+1$-active C_i-number at stage s if it is defined to be a^{i+1}_x for some x so that x is the least member of $[x]_{E_i}$ at stage s. If $i+1 = n$, we also say that x is 0-active.

Lemma 3.25. *Let x and y be distinct i-active numbers at stage s. Then $x \not{E}^s_i\, y$.*

Proof. This is clearly true at stage 0. Suppose $s+1$ is the least stage at which this lemma fails. Let x, Q_1, y, and Q_2 witness this. Let us consider the action at stage s which brought about this situation. At stage s, we must have done more than just defining new parameters, because all new parameters are chosen to be fresh. In particular, if z is fresh, then $[z]_{E^s_i} = \{z\}$, so it cannot contribute to violating our lemma.

There are two parts of the construction at stage s: The action in each of the 4 cases, and then the coding step. We verify that after each of these actions, we have not violated our lemma.

In each of cases 1, 2, and 4, where we can cause a collapse, we have a requirement Q which E_i-collapses one of its parameters z to some other element w. We then declare Q to be inactive. By inductive hypothesis, $[z]_{E_i}$ contains only one i-active number, namely, z. Thus, since z is not i-active at stage $s+1$ since Q becomes inactive, we have added no new active numbers to $[w]_{E_i}$.

In case 3 where $Q = T^i_{j,k}$ where we cause collapse, we collapse k (which may be equivalent to an active number for another requirement) with x, which is $>r$ and not equivalent to a^i_l for any l. Since $x > r$, it is not equivalent to any parameter for a higher priority WP or D requirement. If it is equivalent to the parameter of a higher priority S-requirement, then we undefine the S-requirement's parameter. Similarly, if it is equivalent to a lower priority requirement's parameter, then we undefine this parameter via reinitialization. Either way, we add no new i-active numbers to the E_i-class of k.

Lastly, we have to check that our collapses in the coding step do not cause us to violate this lemma. These are of the form of E_i-collapsing a_x with a_y if we have E_{i-1}-collapsed[1] x with y. We can assume that prior to E_{i-1}-collapsing x with y, both were least in their E_{i-1}-classes. Thus, both a_x and a_y were i-active. It follows that $[a_x]_{E_i}$ and $[a_y]_{E_i}$ had only one i-active element, namely a_x and a_y. But since x and y have collapsed, one has stopped being active. So the newly formed class $[a_x]_{E_i} \cup [a_y]_{E_i}$ still contains only one i-active element. ⊣

LEMMA 3.26. *Let $x = a_k^{i+1}$. Then for every $s > k$, there exists a $j \leq k$ so that $x\ E_{i+1}^s\ a_j^{i+1}$ and a_j^{i+1} is $i+1$-active at stage s.*

PROOF. Let j be the least number in $[k]_{E_i}$. Then $j \leq k$ and a_j^{i+1} is $i+1$-active. By the coding step, $j\ E_i\ k$ implies $a_j^{i+1}\ E_{i+1}\ a_k^{i+1}$. ⊣

LEMMA 3.27. *Let x be a number mentioned before stage s. Suppose that x is not E_i^s-equivalent to any i-active number at stage s. Then for all $t > s$, x is not E_i^t-equivalent to any i-active number at stage t.*

PROOF. Consider the first stage at which x becomes E_i-equivalent to an i-active number. This cannot be caused by our assignment of parameters, since all parameters are assigned to be new. By the same analysis as in Lemma 3.25, any active z which is collapsed with x must simultaneously become inactive. Similarly, this cannot be caused by collapsing for the sake of coding, as this collapses two E_i-classes which already contain i-active numbers by Lemma 3.26. ⊣

COROLLARY 3.28. *For every $i < n$ and $j, k < s$, $j\ E_i^s\ k$ if and only if $a_j^{i+1}\ E_{i+1}^s\ a_k^{i+1}$.*

PROOF. In the Coding stage, we guarantee that $j\ E_i^s\ k$ implies that $a_j^{i+1}\ E_{i+1}^s\ a_k^{i+1}$. To see the reverse, suppose that $j\ \not{E}_i^s\ k$ and let j_0 be least in $[j]_{E_i^s}$ and k_0 be least in $[k]_{E_i^s}$. It follows by construction that $a_{j_0}^{i+1}\ E_{i+1}^s\ a_j^{i+1}$ and $a_{k_0}^{i+1}\ E_{i+1}^s\ a_k^{i+1}$. Then $a_{j_0}^{i+1}$ and $a_{k_0}^{i+1}$ are both active numbers at stage s. It follows by Lemma 3.25 that they cannot be E_{i+1}^s-equivalent. Thus $a_j^{i+1}\ \not{E}_{i+1}^s\ a_k^{i+1}$. ⊣

LEMMA 3.29. *Suppose that x and y are numbers considered before stage s and $x\ \not{E}_i^s\ y$. Suppose that Q is a requirement which is deactivated at stage s (thus all lower priority requirements are reinitialized at stage s). Suppose further that Q is never reinitialized after stage s. Then $x\ \not{E}_i\ y$.*

PROOF.

CASE 1. Neither x nor y is E_i^s-equivalent to any i-active number at stage s. Then by Lemma 3.27, this is true at every $t > s$. Then at any stage $t > s$ we cannot collapse x with y by a WP, D, or S-strategy or during the coding step because neither are equivalent to any active numbers. Lower priority T-requirements will have parameters $r > x, y$. Thus these cannot cause the collapse either. No higher priority T-requirement can cause the collapse as this would reinitialize Q.

CASE 2. Suppose that either x or y is E_i^s-equivalent to an i-active Q-number at stage s for Q a higher priority WP-, D-, or S-requirement. Without loss of generality, we suppose this is true of x. Suppose that $x\ E_i\ y$. Let $t > s$ be the stage at which we cause this collapse. Since Q does not act after stage s, we know

[1] We define $0 - 1 = n - 1$.

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

that x is also E_i^t-equivalent to an i-active Q-number at stage t, and thus cannot be E_i^t-equivalent to any other i-active number by Lemma 3.25. Thus, the collapse must be caused by an active number E^t-equivalent to y or a T-requirement. It cannot be due to a T-requirement by the same reason as in case 1. So we now have three cases to consider:

Case 2a. Suppose that y is not E_i^s-equivalent to an i-active number at stage s. Then by lemma 3.27, this is true at stage t also, so the collapse cannot occur at stage t.

Case 2b. Suppose that y is E_i^s-equivalent to a_j^i for some j. Then by Lemma 3.4, y is E_i^t-equivalent to some i-active a_k^i for some k. Thus, it is not E_i^t-equivalent to a Q'-number at stage t for Q' any WP-, D-, or S-requirement. Since x cannot be equivalent to any a_l^i (since Lemma 3.26 shows that it would then be E_i^t-equivalent to two i-active numbers contradicting Lemma 3.25), the collapse can also not occur due to the coding.

Case 2c. Suppose that y is also E_i^s-equivalent to an i-active Q'-number at stage s for a higher priority WP-, D-, or S-requirement. Then, y is also E_i^t-equivalent to a Q'-number at stage t, and thus cannot be E_i^t-equivalent to any other active number by Lemma 3.25. Thus we cannot cause the collapse at stage t.

Case 3. Suppose that $x\ E_i^s\ a_j^i$ and y is not E_i^s-equivalent to any i-active number. Then Lemma 3.26 shows that $[x]_{E_i^t}$ always contains an i-active number for every $t > s$ and Lemma 3.27 shows that y is never E_i^t-equivalent to an i-active number. Thus x and y can not be E_i^t-equivalent for any $t > s$.

Case 4. Both $x\ E_i^s\ a_j^i$ and $y\ E_i^s\ a_k^i$. Then the only cause of the collapse of $x\ E_i\ y$ is due to the coding step, since neither can ever be E_i^t-equivalent to any other i-active number and T-requirements only use numbers not equivalent to a_l^i for any l. But then we can consider why we collapse $j\ E_{i-1}\ k$. By cases 1–3, the only possibility is that this, in turn was via a coding step. But the coding step at any given stage is finite and originates in a collapse for a WP-, D-, T-, or S-requirement, which we have ruled out in the cases above. ⊣

Lemma 3.30. *Each requirement is satisfied.*

Proof. Suppose towards a contradiction that Q is the highest priority requirement which is not satisfied. Let s be a stage after which Q will not be reinitialized. We consider the cases:

$Q = WP_j^i$: Once the parameter x is chosen after stage s, this is permanent. Either $\varphi_j(x)$ diverges or we collapse $x\ E_i\ \varphi_j(x)$. Either way, the requirement is satisfied.

$Q = D_j^{i,i'}$: Once the parameters $x, x', andz$ are chosen after stage s, this choice is permanent. If $\varphi_j(x)$ or $\varphi_j(x')$ do not converge or no $\varphi_j(y)$ converges to equal z, then φ_j is not a bijection and the requirement is satisfied. Otherwise let t be the first stage we consider Q after these convergences are witnessed. Similarly, if $\varphi_j(x)\ E_{i'}^t\ \varphi_j(x')$, then Lemma 3.29 guarantees that $x \not\mathrel{E_i} x'$ and the requirement is satisfied. Otherwise (possibly after reversing x and x'), we have $\varphi_j(x) = w \not\mathrel{E_{i'}^t} z$. Then we E_i-collapsed x with y. Thus it suffices to show that $w \not\mathrel{E_{i'}} z$. This follows directly from Lemma 3.29.

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

$Q = T^i_{j,k}$: Once the requirement chooses its parameter r after stage s, this is permanent. Suppose W_j intersects infinitely many E_i-classes which do not contain any element of the form a^i_l. Then W_j will enumerate a number x which is not E_i-equivalent to any number a^i_l and also not E_i-equivalent to any number less than r. Thus, once we consider the requirement after such a number x is enumerated into W_j, we will either already have $k \in [W^t_e]_{E_{i,t}}$ or we will collapse x with k.

$Q = S_j$: We have two cases to consider:

Case A. The sequence of parameters $y^0, y^1, \dots$ which are never removed is infinite. In this case, each of these y^l are i-active permanently. It follows that $y^l \not\mathrel{E_i} y^{l'}$ for each pair l, l'. For each one, we have $\{[\varphi_j(y^0)]_{E_0}, [\varphi_j(y^l)]_{E_0}\} = \{[\pi^k y^0]_{E_0}, [\pi^k y^l]_{E_0}\}$ for some k. We next check that this k is the same for each l.

Claim 3.31. *If $k < k'$, then $\pi^k(y^l) \not\mathrel{E_0} \pi^{k'}(z)$ for any z.*

Proof. Suppose otherwise that $\pi^k(y^0)\ E_0\ \pi^{k'}(z)$. Then since π is a reduction of E_0 to itself, we get that $y^0\ E_0\ \pi^{k'-k}(z)$. But $\pi^{k'-k}(z)$ is a number of the form a^n_w for some w. But then y^0 is 0-active and equivalent to another 0-active number of the form of a^n_w by Lemma 3.26, which contradicts Lemma 3.25. ⊣

It follows that for each l we have $\varphi_j(y^l) E_0 \pi^k(y^l)$ for the same number k. Note that we cannot have, for instance, that $\varphi_j(y^0)\ E_0\ \pi^k(y^1)$ and $\varphi_j(y^1)\ E_0\ \pi^k(y^0)$, because then the condition $\{[\varphi_j(y^0)]_{E_0}, [\varphi_j(y^l)]_{E_0}\} = \{[\pi^k y^0]_{E_0}, [\pi^k y^l]_{E_0}\}$ will fail for $l = 2$. By the claim, we have that $\varphi_j(y^l) \notin [\mathrm{im}(\pi^{k+1})]_{E_0}$. Since $y^l \not\mathrel{E_0} y^{l'}$ for each pair, we get that $\pi^k(y^l) \not\mathrel{E_0} \pi^k(y^{l'})$, and we have that $\mathrm{im}(\varphi_j) \cap ([\mathrm{im}(\pi^k)]_{E_0} \smallsetminus [\mathrm{im}(\pi^{k+1})]_{E_0})$ contains infinitely many classes.

Case B. We only have finitely many stable parameters $y_0, y_1, \dots, y^l$. By Lemma 3.22, there is a stage s' after which y^{l+1} is never defined. At a stage $t > s'$, when the requirement is considered, it must either have $\varphi_j(y^l)$ diverge, $\varphi_j(y^0)\ E^t_0\ \varphi_j(y^l)$, in which case it does nothing, but $y^0 \not\mathrel{E_0} y^l$, since both remain 0-active permanently, or it must have $\varphi_j(y^0) \not\mathrel{E^t_0} \varphi_j(y^l)$. In this latter case, we E_0-collapse y^0 with y^l. By Lemma 3.29, we see $\varphi_j(y^0) \not\mathrel{E_0} \varphi_j(y^l)$, so φ_j is not a reduction of E_0 to itself, and the requirement is satisfied. ⊣

Lemma 3.32. *$E_0, \dots, E_{n-1}$ are all equivalent, yet nonisomorphic weakly precomplete ceers. They are all non–self-full.*

Proof. By the requirements C_i, we have $E_0 \leq_c E_1 \leq_c \cdots \leq_c E_n = E_0$, thus they are all equivalent, and by requirements $D^{i,i'}_j$, they are nonisomorphic. By the requirements WP^i_j, they are all weakly precomplete. Since $E_{n-1} \leq_c E_n$ and $E_{n-1} \not\cong E_n$, we see by Lemma 2.3 that the reduction π_n is not onto the classes of E_n. Therefore, the map π is not onto the classes of E_0. But then π is a reduction of E_0 to itself which is not onto the classes of E_0, showing that E_0 is non–self-full. Self-fullness is a property of degrees, so no E_i is self-full. ⊣

Lemma 3.33. *A ceer R is equivalent to E_0 if and only if it is isomorphic to a ceer of the form $E_i \oplus D$ where D is a ceer with either finite or infinite domain and is comprised of finitely many computable classes.*

Proof. Suppose R is of the form $E_i \oplus D$ where D is a ceer comprised of finitely many computable classes. Then $R \equiv_c E_i \oplus \mathrm{Id}_n$ where n is the number of classes in D. But since E_i is non–self-full, we have that $E_i \oplus \mathrm{Id}_n \equiv_c E_i$.

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

Suppose $R \equiv_c E_0$. Then let φ_j be the reduction given by $E_0 \leq_c R \leq_c E_0$, and let ψ be the reduction of R to E_0. Note that $\mathrm{im}(\varphi_j) \subseteq \mathrm{im}(\psi)$. By S_j, for some k, $[\mathrm{im}(\varphi_j)]_{E_0} \cap [\mathrm{im}(\pi^k)]_{E_0} \smallsetminus [\mathrm{im}(\pi^{k+1})]_{E_0}$ is infinite, and, therefore, $K = [\mathrm{im}(\psi)]_{E_0} \cap [\mathrm{im}(\pi^k)]_{E_0} \smallsetminus [\mathrm{im}(\pi^{k+1})]_{E_0}$ is infinite too. Let k be least so that $[\mathrm{im}(\psi)]_{E_0} \cap [\mathrm{im}(\pi^k)]_{E_0} \smallsetminus [\mathrm{im}(\pi^{k+1})]_{E_0}$ is infinite. Then $\mathrm{im}(\psi)$ intersects only finitely many E_0-classes from $\omega \smallsetminus [\mathrm{im}(\pi^k)]_{E_0}$ with witnesses, say, $c_1, c_2, \dots, c_{n_1}$. Since K is the union of the sets $K_m = [\mathrm{im}(\psi)]_{E_0} \cap [\mathrm{im}(\pi^k \circ \pi_n \circ \cdots \circ \pi_{m+1})]_{E_0} \smallsetminus [\mathrm{im}(\pi^k \circ \pi_n \circ \cdots \circ \pi_m)]_{E_0}$, one of them is infinite. Let m be the biggest such number. Then $\mathrm{im}(\psi)$ intersects only finitely many E_0-classes from $[\mathrm{im}(\pi^k)]_{E_0} \smallsetminus [\mathrm{im}(\pi^k \circ \pi_n \circ \cdots \circ \pi_{m+1})]_{E_0}$. Let $d_1, d_2, \dots, d_{n_2}$ be witnesses of these classes.

Now, consider the c.e. set $W = \{i \mid \pi^k \circ \pi_n \circ \cdots \circ \pi_{m+1}(i) \in [\mathrm{im}(\psi)]_{E_0}\}$. Then this W hits infinitely many E_m-classes which are not in the range of π_m. By the T-requirements, it intersects every E_m-class. Thus, $\mathrm{im}(\psi)$ contains $\mathrm{im}(\pi^k \circ \pi_n \circ \cdots \circ \pi_{m+1})$ along with finitely many more classes $[c_j]_{E_0}$, $[d_k]_{E_0}$. Hence, in R, the set of i so that $\psi(i) \in \mathrm{im}(\pi^k \circ \pi_n \circ \cdots \circ \pi_{m+1})$ is c.e. and the set of i so that $\psi(i)$ is in $\bigcup [c_j]_{E_0} \cup \bigcup [d_k]_{E_0}$ is c.e., and this gives a finite partition of ω into c.e. sets. Therefore, these sets are computable. Thus R is equivalent to the uniform join of E_0 restricted to the set $\mathrm{im}(\pi^k \circ \pi_n \circ \cdots \circ \pi_{m+1})$ and a ceer with finitely many computable classes. But E_0 restricted to the set $\mathrm{im}(\pi^k \circ \pi_n \circ \cdots \circ \pi_{m+1})$ is isomorphic to the ceer E_m by Lemma 2.3. ⊣
⊣

Corollary 3.34. *There is a weakly precomplete ceer E so that for any ceer R, $R \equiv_c E$ if and only if R is isomorphic to $E \oplus X$ where X is a c.e. equivalence relation (on a possibly finite universe) comprised of finitely many computable classes.*

Proof. Apply the previous construction with $n = 1$. Note that we have no D-requirements, and thus we have to work slightly harder to ensure non–self-fullness. It suffices to ensure that the class of 0 is not equivalent to any a^1_j, and to do this it suffices by Lemma 3.27 to begin the construction by mentioning the number 0. ⊣

§4. Ceers reducible to one with finite classes.

Theorem 4.1. *The index set of the collection of ceers reducible to one with only finite classes is Σ^0_4-complete.*

Proof. This proof is a standard priority construction using a tree of strategies. This is somewhat unusual in the study of ceers, where most arguments are finite injury arguments.

It is easy to estimate that the desired index set is Σ^0_4. To prove the theorem we fix a Σ^0_4-complete set $S = \{i \mid \exists j W_{g(i,j)} \text{ is coinfinite}\}$ and consider requirements:

$P_{j,k}$: If $[k, \omega) \subseteq \bigcap_{m \leq j} W_{g(i,m)}$, then φ_j is not a reduction of E to a ceer with only finite classes.

$P_{j,k}$-strategy: Step 1. Let x_0 be a fresh number.

Step 2. Wait for $\varphi_j(x_l)$ to converge for every x_l which has been chosen. If φ_j is injective on the set of chosen x_l, then go to Step 3. Otherwise, we will have x_{l_1}, x_{l_2} so that we keep $x_{l_1} \not\!\!E\, x_{l_2}$, yet $\varphi_j(x_{l_1}) = \varphi_j(x_{l_2})$, showing that φ_j is not a reduction of E to any ceer.

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

STEP 3. Collapse each defined x_l to be E-equivalent to x_0. Let n be least so x_n is not yet defined. Choose x_n to be fresh and go back to Step 2.

We put these strategies on a tree as usual for an infinite-injury construction. We will omit some details of the construction in favor of clarity. We fix a tree of strategies with nodes $\{\infty, f\}^{<\omega}$. Each strategy on level $\langle j, k\rangle$ of the tree will be a $P_{j,k}$-strategy. If a node β is a $P_{j,k}$-strategy, and $\beta \succeq \alpha\infty$ where α is a $P_{j,k'}$-strategy, then we say β is redundant and it never acts.

When visited, a $P_{j,k}$-strategy will have outcomes $\infty < f$. A stage is expansionary for β, a $P_{j,k}$-node if the least element of $[k, \omega) \smallsetminus \bigcap_{m\leq j} W_{g(i,m)}$ is larger at the current stage then at the last stage when β was visited. β only acts on stages where β is visited which are expansionary for β. If it acts and it goes to Step 3 (thus choosing a new x_n), it will take the outcome ∞. Otherwise, it takes the outcome f. As usual, we define the current path by the outcomes taken by nodes visited, and if we visit a node left of β, then we reinitialize β. This concludes the description of the construction.

Note that since the strategies each work with fresh elements x_i and only collapse them to other elements chosen by that strategy, if α chooses x_l and x_n and does not choose to collapse x_l with x_n, then $x_l \not\mathrel{E} x_n$.

LEMMA 4.2. *Suppose that for every j, $W_{g(i,j)}$ is cofinite. Then no φ_j is a reduction of E to a ceer whose classes are finite.*

PROOF. Fix j and let k be least so that $[k, \omega) \subseteq \bigcap_{m\leq j} W_{g(i,m)}$. Let tp be the true path, let α be the $P_{j,k}$-strategy on tp. Let s be a stage large enough that no node left of α is visited after stage s. Thus, at any α-expansionary stage $t > s$, we define x_0 to have its final value. We consider two cases: $\alpha\infty \preceq$ tp or $\alpha f \preceq$ tp. In the first case, φ_i is injective on the set $\{x_l \mid l \in \omega\}$, but we make each of these E-equivalent. Suppose $E \leq_c R$ is witnessed by the reduction φ_j. Then $\{\varphi_j(x_m) \mid m \in \omega\}$ either defines more than one R-class, in which case φ_j is not a reduction of E to R, or it defines an infinite subset of one class, showing that R has an infinite class.

In the second case, the strategy gets stuck in Step 2: This means that either φ_j is not total or for some $l < k$, we have $\varphi_j(x_l) = \varphi_j(x_k)$, but we never collapse x_l and x_k. Thus, φ_j is not a reduction of E to any ceer. ⊣

LEMMA 4.3. *Suppose that for some j, $W_{g(i,j)}$ is coinfinite. Then E is reducible to a ceer with finite classes.*

PROOF. Let j be least so that $W_{g(i,j)}$ is coinfinite. Then for every $j' \geq j$ and any k', any $P_{j',k'}$-strategy has only finitely many expansionary stages. Thus, each strategy can only create finite classes. Thus, the only strategies which can create infinite classes in E are the strategies on the true path which are $P_{j',k'}$-strategies with $j' < j$. But there are only finitely many of these which are not redundant—at most one for each $j' < j$. Thus E is a ceer with at most j infinite classes. But note that each of these classes are computable: The strategy chooses $x_0 < x_1 < x_2 < \cdots$ and this forms the class. We can let R be the ceer formed by replacing each of these classes by a single point. It is easy to see that $E \leq_c R$, and R has only finite classes. ⊣

⊣

Gao and Gerdes [11] gave an indirect proof that there is a ceer E all of whose classes are computable, but E is not reducible to any ceer with only finite classes. They do this by showing that the index set of ceers with all computable classes is

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

Π^0_4-complete, but the index set of ceers reducible to one with only finite classes is Σ^0_4 (they only show Π^0_3-hardness). They ask for a direct construction of such a ceer.

OBSERVATION 4.4. *The proof of Theorem* 4.1 *gives a direct construction of a ceer with computable classes which does not reduce to one with only finite classes.*

PROOF. For any $i \notin S$, the ceer E produced in the previous construction has computable classes (again, since the classes are chosen as $x_0 < x_1 < x_2 < \cdots$), but does not reduce to a ceer with only finite classes. ⊣

§5. Strong minimal covers of sets of degrees of ceers. We now turn our attention to some questions about least or minimal upper bounds for some subsets in the structure of ceers. Gao and Gerdes [11] asked whether Id' is a least upper bound of $\{\mathrm{Id}'_k \mid k \in \omega\}$ and Andrews and Sorbi [4] asked whether there is a minimal upper bound for the set $\{\mathrm{Id}^{(n)} \mid n \in \omega\}$.

DEFINITION 5.1. If S is a subset of a preodered set $\langle P, \leqslant\rangle$, we say that $c \in P$ is a strong minimal cover of S if $c \notin S$ and for every $x \in P$, $x \leqslant c \iff$ either $x \equiv c$ or $\exists y \in S\,(x \leqslant y)$.

As usual, we write shortly $x \equiv y$ if $x \leqslant y \,\&\, y \leqslant x$ and write $x < y$ if $x \leqslant y \,\&\, y \nleqslant x$.

Obviously, every strong minimal cover of S is an upper bound for S. We will deal with *internally unbounded* subsets of a preodered set P, i.e., subsets S that have a following property: $\forall x \in S \exists y \in S (x < y)$. For instance, the sets $\{\mathrm{Id}_n \mid n \in \omega\}$ and $\{\mathrm{Id}^{(n)} \mid n \in \omega\}$ are internally unbounded.

LEMMA 5.2. (i) *Let S be a subset of a preodered set $\langle P, \leqslant\rangle$ that has a least upper bound b and a strong minimal cover c. If $b \in S$ then $b < c$, otherwise, $c \equiv b$.*

(ii) *If S is internally unbounded set then*

- *a strong minimal cover of S is a minimal upper bound of S;*
- *if S has two incomparable strong minimal covers then S has no least upper bound.*

Proof is obvious.

THEOREM 5.3. *Let (E_i) be a uniform c.e. sequence of nonuniversal ceers. Then $\{\oplus_{i \leq j} E_i \mid j \in \omega\}$ has infinitely many incomparable strong minimal covers.*

PROOF. We build infinitely many ceers R_k. Throughout the construction, we will have some columns of R_k reserved for coding. If we reserve the jth column of R_k as a coding column for E_i, then for every x, y, we ensure that $\langle j, x\rangle \; R_k \; \langle j, y\rangle$ if and only if $x \; E_i \; y$. We say that a column is destroyed if every number in the column is equivalent to a number in a smaller column. We will ensure that if x, y are in different coding columns, then $x \not\mathrel{R_k} y$.

We construct $(R_k)_{k \in \omega}$ to satisfy the following requirements:

$C_{n,k}$: There is an R_k-coding column for E_n, i.e., there is a j so that $\forall x, y(\langle j, x\rangle \; R_k \; \langle j, y\rangle \leftrightarrow x \; E_n \; y)$

$P_{i,j,k}$: If W_i intersects the closures of infinitely many nondestroyed columns in R_k, then W_i intersects $[j]_{R_k}$.

$D_{e,k,k'}$: The function φ_e does not give a reduction of R_k to $R_{k'}$ if $k \neq k'$.

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

For convenience, we choose to build each R_k so that the first column of R_k is exactly E_0 and the second column is exactly E_1. These two columns are coding columns, and this is not subject to injury.

C_n-strategies. Pick a new column and decide to code E_n into this column. Restrain this column from being destroyed by a lower priority $P_{i,j}$-requirement.

$P_{i,j}$-strategies. Wait for some x to enter W_i which is in a column $> j$ which is not restrained by any higher priority strategy from being destroyed. At this point, collapse the entire column of x to be equivalent to j. We say this column has been destroyed.

$D_{e,k,k'}$-strategies. We first pick a new column j of R_k. As long as it appears that φ_i gives a reduction of R_k into restrained columns of $R_{k'}$, we will threaten to code a universal ceer on this column of R_k. We will argue below that we do not succeed in this coding (in brief, this is because the restrained $R_{k'}$-columns will together be equivalent to a finite uniform join of the nonuniversal E_i's, but the universal degree is uniform-join irreducible [5]), but the threat will suffice to guarantee that φ_e is not a reduction of R_k into the restrained columns of $R_{k'}$. If the image of φ_e contains two classes in nonrestrained columns, then we will explicitly diagonalize. We fix T a universal ceer.

Step 1. We use a parameter n, which begins with $n = 1$. We choose $a_0 = \langle j, 0\rangle$ and $a_1 = \langle j, 1\rangle$. If we see a stage s so that $\{a_i \mid i \leq n\} \subseteq \mathrm{domain}(\varphi_e^s)$ and for each $x, y \leq n$, $a_x\ R_k\ a_y \leftrightarrow \varphi_e(a_x)\ R_{k'}\ \varphi_e(a_y)$, then we R_k-collapse each pair a_x, a_y with $x, y \leq n$ so that $x\ T^s\ y$. We choose the least element of the jth column which is not R_k^s equivalent to any a_i with $i \leq n$, and let this be a_{n+1}, we increment $n = n + 1$. While we wait for these convergences and equivalences, at stage s, we collapse any element of $\{\langle j, i\rangle \mid i \leq s\} \setminus \{a_i \mid i \leq n\}$ with a_0 (while doing this, we do not say D is acting, and we do not reinitialize lower priority requirements – we do that when we increment n or act as in Step 2). If, for some x, y, $a_x \not R_k^s\ a_y$, $\varphi_e(a_x)$ and $\varphi_e(a_y)$ converge and are not in columns of $R_{k'}$, which are restrained by higher priority requirements, then we go to Step 2.

Step 2. If $\varphi_e(a_x)\ R_{k'}^s\ \varphi_e(a_y)$, then we destroy the jth column of R_k by making every element of $[a_x]_{R_k^s}$ equivalent to $\langle 0, 0\rangle$ and every other element equivalent to $\langle 1, 0\rangle$.

Otherwise, we destroy the jth column of R_k by making every element equivalent to $\langle 0, 0\rangle$. In addition, we will destroy the columns of $\varphi_e(a_x)$ and $\varphi_e(a_y)$ in $R_{k'}$ as follows: If $\varphi_e(a_x)$ and $\varphi_e(a_y)$ are in different columns of $R_{k'}$, then destroy the column of $\varphi_e(a_x)$ by making every element equivalent to $\langle 0, 0\rangle$ and destroy the column of $\varphi_e(a_y)$ by making every element equivalent to $\langle 1, 0\rangle$. Lastly, if $\varphi_e(a_x)$ and $\varphi_e(a_y)$ are in the same column of $R_{k'}$, then we destroy the column by making every element in $[\varphi_e(a_x)]_{R_{k'}^s}$ equivalent to $\langle 0, 0\rangle$ and every other number in the column equivalent to $\langle 1, 0\rangle$.

Construction. We construct a supplementary partial computable function $d(e, k, k', s)$ along with building the ceers R_k. We fix an ordering of all requirements in order type ω and fix a computable correspondence between the stages of the construction and the requirements so that: at any stage $s + 1$ of the construction, we deal exactly with one of the requirements; every requirement is considered at infinitely many stages.

We fix a universal ceer T and its approximation T^s. Let E_n^s be a computable approximation of a uniformly c.e. sequence E_n of nonuniversal ceers.

Reinitializing a $C_{n,k}$-strategy means that a coding column of equivalence E_n in R_k, if any, is stopped being a coding column. Reinitializing a $D_{e,k,k'}$-strategy at stage $s+1$ means to set $d(e,k,k',s+1)$ undefined. Reinitializing a $P_{i,j,k}$-strategy means doing nothing.

Stage 0. Initialize all requirements. For every k, set $R_k^0 = Id$.

Stage $s+1$ for $C_{n,k}$-requirement with $n > 1$. If R_k has no chosen coding column for E_n, then pick a new, say jth, column in R_k with $j > 1$ and declare this column to be the chosen coding column of E_n in R_k. Restrain the jth column of R_k from being destroyed by a lower priority requirement.

Correct all coding: for every n', k', if the j'-th column of $R_{k'}$ is a coding column of $E_{n'}$, and $x\ E_{n'}^{s+1}\ y$, then $R_{k'}$-collapse $\langle j', x\rangle$ with $\langle j', y\rangle$.

STAGE $s+1$ FOR $P_{i,j,k}$-REQUIREMENT. If the equivalence class $[j]_{R_k^s}$ does not intersect W_i^s and there are x and $j' > 1$ so that $\langle j', x\rangle \in [W_i^s]_{R_k^s}$, $j < j'$, and the j'-th column in R_k is not yet destroyed and not restrained by a strategy of higher priority from being destroyed, then R_k-collapse j with the entire j'-th column. This means that we add $\langle j, \langle j', z\rangle\rangle$ into the computable equivalence relation R_k^{s+1} for every z. We say that the j'-th column in R_k has been destroyed. Reinitialize all strategies of lower priority.

STAGE $s+1$ FOR $D_{e,k,k'}$-REQUIREMENT. We distinguish the following three cases.

CASE 1. If $d(e,k,k',s)$ is not defined, then pick a new column $j > 1$ in R_k and define $d(e,k,k',s+1) = \langle j, 1\rangle$, denote $\langle j, 0\rangle$ and $\langle j, 1\rangle$ by a_0 and a_1. Restrain the jth column of R_k from being destroyed by a lower priority requirement. Reinitialize all strategies of lower priority.

CASE 2. If $d(e,k,k',s)$ is defined and equals $\langle j, n\rangle$, $n \geq 1$, $\{a_i \mid i \leq n\} \subseteq \mathrm{domain}(\varphi_e^s)$ and, for each $x, y \leq n$, $a_x\ R_k^s\ a_y \leftrightarrow \varphi_e(a_x)\ R_{k'}^s\ \varphi_e(a_y)$, then check whether there exist x, y so that:

(1) $0 \leq x < y \leq n$,
(2) $a_x \not{R}_k^s\ a_y$,
(3) $\langle \varphi_e(a_x)\rangle_0 > 1$ and $\langle \varphi_e(a_y)\rangle_0 > 1$,
(4) $\varphi_e(a_x)$ and $\varphi_e(a_y)$ are not in columns of $R_{k'}$ which are restrained by higher priority requirements from being destroyed.

If so, go to Subcase 2.1, otherwise, go to Subcase 2.2.

Subcase 2.1.

(i) If $\varphi_e(a_x)\ R_{k'}^s\ \varphi_e(a_y)$, then destroy the jth column of R_k by R_k-collapsing every element of $[a_x]_{R_k^s}$ with $\langle 0, 0\rangle$ and every other element of the jth column with $\langle 1, 0\rangle$. Define $d(e,k,k',s+1) = \langle 0, 0\rangle$. Reinitialize all strategies of lower priority.

(ii) If $\varphi_e(a_x) \not{R}_{k'}^s\ \varphi_e(a_y)$ and $\varphi_e(a_x)$ and $\varphi_e(a_y)$ are in different columns of $R_{k'}$, then destroy the jth column of R_k by R_k-collapsing every element of this column with $\langle 0, 0\rangle$; destroy the column of $\varphi_e(a_x)$ in $R_{k'}$ by $R_{k'}$-collapsing every element of the column with $\langle 0, 0\rangle$; and destroy the column of $\varphi_e(a_y)$ by $R_{k'}$-collapsing every element of the column with $\langle 1, 0\rangle$. Define $d(e,k,k',s+1) = \langle 0, 0\rangle$. Reinitialize all strategies of lower priority.

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

(iii) If $\varphi_e(a_x) \not R^s_{k'} \varphi_e(a_y)$ and $\varphi_e(a_x)$ and $\varphi_e(a_y)$ are in the same column of $R_{k'}$, then destroy the jth column of R_k by R_k-collapsing every element of this column with $\langle 0, 0\rangle$; destroy the column of $\varphi_e(a_x)$ in $R_{k'}$ by $R_{k'}$-collapsing every element of $[\varphi_e(a_x)]_{R^s_{k'}}$ with $\langle 0, 0\rangle$ and every other element in the column with $\langle 1, 0\rangle$. Define $d(e, k, k', s+1) = \langle 0, 0\rangle$. Reinitialize all strategies of lower priority.

Subcase 2.2.

(i) R_k-collapse each pair a_x, a_y with $x, y \leq n$ so that $x \; T^s \; y$;
(ii) choose the least $\langle j, a\rangle$ which is not yet R_k-equivalent to any a_i with $i \leq n$, define $d(e, k, k', s+1) = a_{n+1} = \langle j, a\rangle$.

CASE 3. If Cases 1,2 do not hold, then R_k-collapse any element of $\{\langle j, i\rangle \mid i \leq s\} \setminus \{a_i \mid i \leq n\}$ with a_0.

End of stage $s+1$. For each k perform the symmetrical and transitive closure of the set of pairs that have been enumerated into R_k by the end stage $s+1$. Go to the next stage.

Verification. The verification is done via the following lemmas. We say that a column is an active column at stage s if it has not been destroyed by stage s. Obviously, columns 0 and 1 are active at any stage.

LEMMA 5.4. *There is no stage s and numbers x and y in different active columns at stage s so that $x \; R^s_k \; y$.*

PROOF. We collapse elements within the same column at stages for C-requirements to correct coding or in Subcase 2.2 or 3 at stages for D-requirements followed by performing End of stage. These collapses are made without destroying columns. During other collapsing pairs of numbers of different columns, at least one of these columns is destroyed, thus no longer is an active column. ⊣

We say that a column of R_k is a permanent coding column if it is chosen to code some E_n into R_k by a $C_{n,k}$ strategy which is never reinitialized after this choice.

LEMMA 5.5. *If x and y are in different permanent coding columns of R_k, then $x \not R_k \; y$.*

PROOF. From some stage onwards, both x and y are in active columns at stage s. Thus by the previous lemma, $x \not R^s_k \; y$ for each s, showing that $x \not R \; y$. ⊣

LEMMA 5.6. *At any stage s, if x is in the jth column, and the jth column has been destroyed, then x is equivalent to an element in an active column.*

PROOF. When we destroy a column, we make every element equivalent to an element in a smaller column. Either this column is active, or it is destroyed, making every element equivalent to an element in a smaller column. Since ω is well ordered, and 0 and 1 are coding columns in each R_k, we see that x is equivalent to an element in an active column. ⊣

LEMMA 5.7. *Each strategy reinitializes lower priority strategies only finitely often. For every e, k, k', $\lim_{s\to\infty} d(e, k, k', s)$ is finite. And each requirement is satisfied.*

PROOF. We show the result by induction on the priority of requirements, identifying the claim that $\lim_{s\to\infty} d(e, k, k', s)$ is finite with the requirement $D_{e,k,k'}$. So, we may assume that every strategy of higher priority than $\mathcal{S}$ reinitializes lower priority requirements only finitely often, is satisfied, and all higher priority $D_{e,k,k'}$-requirements have $\lim_{s\to\infty} d(e,k,k',s)$ finite.

Let s_0 be the least stage of the construction so that after stage s_0 each strategy of higher priority than $\mathcal{S}$ does not reinitialize lower priority strategies after stage s_0. If $\mathcal{S}$ is a $C_{n,k}$-strategy, then it never reinitializes lower priority strategies and once it chooses a coding column after stage s_0, this choice of column is permanent, and it succeeds in coding E_n into this column of R_k. If $\mathcal{S}$ is $P_{i,j,k}$-strategy, note that it can reinitialize lower priority strategies at most once after stage s_0. Note that since each higher priority strategy only restrains at most one column of R_k, if W_i intersects the closures of infinitely many nondestroyed columns of R_k, then it will intersect the closure of one which is not restrained by a higher priority requirement, and $\mathcal{S}$ will act satisfying the requirement.

Lastly, we consider the case that $\mathcal{S}$ is a $D_{e,k,k'}$-strategy. A value of the function $\lambda s d(e,k,k',s)$ may be undefined or be any natural number. Note that $D_{e,k,k'}$-strategy reinitializes lower priority strategies only by Subcase 2.1 and if it did so at stage $s_1+1 > s_0$ then $d(e,k,k',s_1)$ equals to some number $\langle j,n\rangle$ with $n \geq 1$ while $d(e,k,k',s_1+1) = \langle 0,0\rangle$. Besides, $d(e,k,k',s) = \langle 0,0\rangle$ for all $s \geq s_1+1$. Therefore, $D_{e,k,k'}$-strategy does not reinitialize lower priority strategies after stage s_1+1, since only Case 3 holds at these stages. Note that if we enter Subcase 2.1, we explicitly diagonalize to ensure that $\mathcal{S}$ is satisfied. This only uses that $\langle 0,0\rangle \not R_k \langle 1,0\rangle$ for each k. Thus, we have the result if the strategy ever enters Subcase 2.1 after stage s_0.

Now, we prove that the $D_{e,k,k'}$-requirement is satisfied and $\lim_{s\to\infty} d(e,k,k',s)$ is finite, given that it is not reinitialized after stage s_0 and that after stage s_0, $\mathcal{S}$ never enters Subcase 2.1. Since $d(e,k,k',s)$ can't be undefined in all stages after the stage s_0, we can assume that $d(e,k,k',s_0)$ is defined due to Case 1. So, $d(e,k,k',s)$ is defined and is different from $\langle 0,0\rangle$ and Subcase 2.1 does not hold for every $s \geq s_0$. Suppose towards a contradiction that $\lim_{s\to\infty} d(e,k,k',s)$ is infinite. Then Subcase 2.2 holds infinitely often and φ_e reduces a universal ceer T to the closures of finitely many columns of $R_{k'}$. Each of these are restrained from being destroyed by a higher priority requirement. Since every higher priority $D_{e',l,l'}$-requirement has $\lim_{s\to\infty} d(e',l,l',s)$ finite, its column is either destroyed or contains only finitely many classes via cofinitely being in Case 3. Thus, each of these columns in $R_{k'}$ is either destroyed, or is a permanent coding column, or has only finitely many classes. Thus $T \leq_c \oplus_{i\leq m} E_i \oplus \mathrm{Id}_m$ for some m. But since the universal degree is uniform-join irreducible [5], we must have that $T \leq_c E_i$ for some i, but this contradicts each E_i being nonuniversal.

Thus we know that $\lim_{s\to\infty} d(e,k,k',s)$ is finite. Then we must always take Case 3 after some $s_1 > s_0$. Therefore, either φ_e is not a total function or the equivalence $a_x \; R_k \; a_y \leftrightarrow \varphi_e(a_x) \; R_{k'} \; \varphi_e(a_y)$ fails for some $x, y \leq \lim_{s\to\infty}\langle d(e,k,k',s)\rangle_1$, and the strategy succeeds. Note that the R_k-closure of $\langle d(e,k,k',s)\rangle_0$ column consists of finitely many equivalence classes. ⊣

Lemma 5.8. *Each R_k is a strong minimal cover for $\{\oplus_{i\leq j} E_i \mid j \in \omega\}$.*

Proof. Since each E_i is coded into some column of R_k and these columns have disjoint R_k-classes, we see that every ceer in $\{\oplus_{i\leq j} E_i \mid j\in\omega\}$ is reducible to R_k. Now, suppose $X \leq_c R_k$ via a computable function f. Let W_i be the image of f.

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press

If it intersects the closures of only finitely many nondestroyed columns, then we can reduce X to the uniform join of the finitely many E_i or finite ceers coded on these columns. Thus $X \leq_c \oplus_{i\leq m} E_i \oplus \mathrm{Id}_n$ for some m, n. But then $X \leq_c \oplus_{i\leq m+n} E_i$. Otherwise, W_i intersects the closures of infinitely many nondestroyed columns. Then, by the $P_{i,j}$-requirements, W_i intersects every class. Since the reduction $X \leq_c R_k$ is onto the classes of R_k, we have that $X \equiv R_k$. ⊣

⊣

We apply Theorem 5.3 combined with Lemma 5.2 to get several corollaries below. To prove them we need to show that suitable sets of ceers are of the form covered by Theorem 5.3 and are internally unbounded.

Note that our first corollary provides another proof that there are infinitely many incomparable dark minimal ceers as in [5, Theorem 3.3].

Corollary 5.9. *There are infinitely many strong minimal covers for* $\{\mathrm{Id}_n \mid n \in \omega\}$ *and no least upper bound.*

Proof. We show that the $\leq_c$-downward closure of $\{\mathrm{Id}_n \mid n \in \omega\}$ is the same as the $\leq_c$-downward closure of $\{\oplus_{i\leq n} \mathrm{Id}_i \mid n \in \omega\}$. The former is clearly internally unbounded, so we can apply Theorem 5.3 and Lemma 5.2 to yield the result.

To see that these two downward closures are equal, it suffices to see that the former is closed under uniform-join, which follows from $\mathrm{Id}_n \oplus \mathrm{Id}_m \equiv_c \mathrm{Id}_{n+m}$. ⊣

Corollary 5.10. *There are infinitely many minimal upper bounds for* $\{\mathrm{Id}'_n \mid n \in \omega\}$ *and no least upper bound.*

Proof. Similarly, we need only show that the downward-closure of this collection is closed under uniform join. It is not difficult to see that $\mathrm{Id}'_n \oplus \mathrm{Id}'_m \leq \mathrm{Id}'_{n+m}$. In fact, $E' \oplus R' \leq (E \oplus R)'$ holds for all ceers by [4, Lemma 2.3] ⊣

Observation 5.11. *In fact,* Id' *is also not a minimal upper bound of* $\{\mathrm{Id}'_n \mid n \in \omega\}$.

Proof. Consider the ceer $\oplus_{n\in\omega} \mathrm{Id}'_n$. This is an upper bound of $\{\mathrm{Id}'_n \mid n \in \omega\}$, and a direct reduction shows that it reduces to Id'. But since jumps are uniform join irreducible [4], this is strictly below Id'. ⊣

The following answers a question from [4]:

Corollary 5.12. *There are infinitely many strong minimal covers and no least upper bound to the set* $\{\mathrm{Id}^{(n)} \mid n \in \omega\}$.

Proof. Again, it suffices to show that for $n \leq m$, $\mathrm{Id}^{(n)} \oplus \mathrm{Id}^{(m)} \leq_c \mathrm{Id}^{(m)}$. This is true due to [4, Lemma 2.3] that states: $R' \oplus S' \leq_c (R \oplus S)'$ for any ceers R, S. Then by induction, we get: $R^{(n)} \oplus S^{(n)} \leq_c (R \oplus S)^{(n)}$ for any n. Therefore

$$\mathrm{Id}^{(n)} \oplus \mathrm{Id}^{(m)} \leq_c \mathrm{Id}^{(m)} \oplus \mathrm{Id}^{(m)} \leq_c (\mathrm{Id} \oplus \mathrm{Id})^{(m)} = \mathrm{Id}^{(m)}.$$

⊣

§6. **Observations on minimal ceers.** Gao and Gerdes [11] showed that $\mathrm{Id}_1 <_c \mathrm{Id}_2 <_c \mathrm{Id}_3 <_c \cdots <_c \mathrm{Id}$ and for every $n > 1$ and any ceer R with infinitely many classes, $\mathrm{Id}_n <_c R$. Furthermore, every ceer with finitely many classes is $\equiv_c$ to one of Id_n. This implies that, when we examine the notion of a minimal ceer, we should consider minimality within the collection of ceers with infinitely many classes.

Definition 6.1. We call a ceer R with infinitely many classes to be minimal if, for every ceer S, if $S \leq_c R$ and S has infinitely many classes, then $S \equiv_c R$.

Id is the natural example of a minimal ceer. The following minimality criterion was used by Andrews and Sorbi [5] to construct minimal ceers, but it was not known to be an equivalence. We here show the other implication.

Theorem 6.2. *A ceer R with infinitely many classes is minimal if and only if $R \equiv_c$ Id or, for every c.e. set W, if W hits infinitely many R-classes then it hits every R-class.*

Proof. Let R be any minimal ceer and let a c.e. set W hit infinitely many but not all R-classes. Let us show that $R \equiv_c Id$. Pick a number a so that $[a]_R \cap W = \emptyset$. We choose a computable function f with range W and define a seer S by $x \mathrel{S} y \iff f(x) \mathrel{R} f(y)$. Then $S \leq_c R$ via the function f. Since R is minimal and S has infinitely many classes it follows that $x \mathrel{R} y \iff g(x) \mathrel{S} g(y)$ for some computable function g and all x, y. Note that $a \in \omega \smallsetminus [\mathrm{im}(f \circ g)]_R$, and, therefore, $a \not\mathrel{R} (f \circ g)(a)$. This immediately implies that $a, (f \circ g)(a), (f \circ g)^{(2)}(a)$ are pairwise nonequivalent relative to R. By iterating the function $f \circ g$ on a, we obtain an infinite c.e. sequence of numbers lying in distinct R-classes. If h computably enumerates this sequence then h defines a reduction Id $\leq_c R$. Therefore, Id $\equiv_c R$ by minimality of R.

Suppose now that $R \equiv_c$ Id or, for every c.e. set W, if W hits infinitely many R-classes then it hits each of them. If $R \equiv_c$ Id, then we have nothing to prove, so we suppose we are in the latter case. If a ceer S has infinitely many classes and $S \leq_c R$ via some computable function f, then the range W of f hits infinitely many R-classes, and, therefore im(f) hits each R-class, i.e., $S \leq_c R$ is an onto-reduction. Hence, $R \leq_c S$. ⊣

Theorem 6.3. *There is an infinite $\leq_c$ anti-chain of weakly precomplete minimal ceers.*

Proof. Andrews and Sorbi [5, Theorem 3.3] showed that there are infinitely many incomparable minimal dark ceers. They proceed to build these ceers E_i for $i \in \omega$ via a finite injury argument where each requirement may cause some collapses (respecting the restraint placed by higher priority requirements) and may place a finite restraint, i.e., there are finitely many triples (a, b, j) which represent that a and b are restrained from becoming E_j-collapsed by lower priority requirements.

We need only note that we can add a requirement of type WP^i_j to ensure that E_i has a φ_j-fixed point within this framework. These requirements need to place no restraint, and they obey such restraints as long as the witness x chosen is distinct from any of the restrained classes. ⊣

Acknowledgments. The first author's research was partially supported by NSF grant DMS-160022 and by Grant 3952/GF4 of the Science Committee of the Republic of Kazakhstan. The second author's research was partially supported by binational NSF grant DMS-1600625 entitled "Collaboration in Computability" and by Grant AP05131579 of the Science Committee of the Republic of Kazakhstan.

REFERENCES

[1] U. Andrews, S. Badaev, and A. Sorbi, *A survey on universal computably enumerable equivalence relations*. ***Computability and Complexity: Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday*** (A. Day, M. Fellows, N. Greenberg, B. Khoussainov, A. Melnikov, and F. Rosamond, editors), Lecture Notes in Computer Science, vol. 10010, Springer, Berlin, 2017, pp. 418–451.

[2] U. Andrews, S. Lempp, J. S. Miller, K. M. Ng, L. San Mauro, and A. Sorbi, *Universal computably enumerable equivalence relations*, this Journal, vol. 79 (2014), no. 1, pp. 60–88.

[3] U. Andrews and A. Sorbi, *The complexity of index sets of classes of computably enumerable equivalence relations*, this Journal, vol. 81 (2016), no. 4, pp. 1–21.

[4] ———, *Jumps of computably enumerable equivalence relations*. ***Annals of Pure and Applied Logic***, vol. 169 (2018), no. 3, pp. 243–259.

[5] ———, *Joins and meets in the structure of ceers*. ***Computability***, vol. 8 (2019), no. 3–4, pp. 193–241.

[6] S. Badaev and A. Sorbi, *Weakly precomplete computably enumerable equivalence relations*. ***Mathematical Logic Quarterly***, vol. 62 (2016), pp. 111–127.

[7] Y. L. Ershov, *Positive equivalences*. ***Algebra and Logic***, vol. 10 (1973), no. 6, pp. 378–394.

[8] ———, ***Theory of Numberings***, Nauka, Moscow, 1977. (Russian).

[9] ———, *Theory of numberings*, ***Handbook of Computability Theory*** (E. G. Griffor, editor), Studies in Logic and the Foundations of Mathematics, vol. 140, North-Holland, Amsterdam, 1999, pp. 473–503.

[10] E. Fokina, B. Khoussainov, P. Semukhin, and D. Turesky. *Linear orders realised by c.e. equivalence relations*, this Journal, vo. 81 (2016), no. 2, pp. 463–482.

[11] S. Gao and P. Gerdes, *Computably enumerable equivalence realations*. ***Studia Logica***, vol. 67 (2001), pp. 27–59.

[12] A. Gavryushkin, B. Khoussainov, and F. Stephan, *Reducibilities among equivalence relations induced by recursively enumerable structures*. ***Theoretical Computer Science***, vol. 612 (2016), pp. 137–152.

[13] A. Gavryushkin, S. Jain, B. Khoussainov, and F. Stephan, *Graphs realised by r.e. equivalence relations*. ***Annals of Pure and Applied Logic***, vol.165 (2014), no. (7–8), pp. 1263–1290.

[14] B. Khoussainov, *A journey to computably enumerable structures* (*Tutorial Lectures*), ***Sailing Routes in the World of Computation*** (F. Manea, R. G. Miller, and D. Nowotka, editors), Springer, New York, 2018, pp. 1–19.

[15] B. Khoussainov and A. Miasnikov, *Finitely presented expansions of groups, semigroups, and algebras*. ***Transactions of the American Mathematical Society***, vol. 366 (2014), no. 3, pp. 1455–1474.

[16] A. H. Lachlan, *A note on positive equivalence relations*. ***Zeitschrift für Mathematische Logik und Grundlagen der Mathematik***, vol. 33 (1987), pp. 43–46.

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

[18] R. I. Soare, ***Recursively Enumerable Sets and Degrees***, Perspectives in Mathematical Logic, Omega Series, Springer-Verlag, Heidelberg, 1987.

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF WISCONSIN
MADISON, WI 53706-1388, USA
E-mail: andrews@math.wisc.edu
URL: http://www.math.wisc.edu/∼andrews/

and

DEPARTMENT OF FUNDAMENTAL MATHEMATICS
AL-FARABI KAZAKH NATIONAL UNIVERSITY
ALMATY 050040, KAZAKHSTAN
E-mail: serikzhan.badaev@kaznu.kz

https://doi.org/10.1017/jsl.2019.39 Published online by Cambridge University Press