



Cooper, S. B., Gay, J., Harris, C., Lee, K. I., & Morphett, A. (2018).
Linearisations and the Ershov Hierarchy. *Computability*, 7(2-3), 143-169.
<https://doi.org/10.3233/COM-170080>

Peer reviewed version

Link to published version (if available):
[10.3233/COM-170080](https://doi.org/10.3233/COM-170080)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via IOS Press at <https://content.iospress.com/articles/computability/com080>. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/pure/about/ebr-terms>

LINEARISATIONS AND THE ERSHOV HIERARCHY

S. BARRY COOPER, JAMES GAY, CHARLES M. HARRIS, KYUNG IL LEE,
AND ANTHONY MORPHETT

ABSTRACT. A partial order is computably well founded if it does not computably embed a copy of ω^* , the order type of the negative integers. It is computably scattered if it does not computably embed a copy of η , the order type of \mathbb{Q} . It is known that, for each of these properties, there are computable partial orders satisfying the property which do not have a computable linear extension with the same property. Rosenstein showed, however, that for both of these properties, every computable partial order satisfying the property has a Δ_2^0 linear extension also satisfying the property. Thus, linear extensions of a computable order preserving the properties of computable well foundedness or computable scatteredness can always be found at the Δ_2^0 level of the arithmetical hierarchy, but not at the Δ_1^0 level. In this paper, we investigate at which level of the Ershov hierarchy such linear extensions can be found. We show that, for both properties, every computable partial order satisfying the property has an ω -c.e. linear extension with the same property. We establish that this is the best possible result within the Ershov hierarchy by constructing, respectively, computably well founded and computably scattered orders which do not have n -c.e. linear extensions which are computably well founded and computably scattered respectively, for any $n < \omega$. In a strengthening of Rosenstein's theorems in another direction, we show that a linear extension preserving each of these properties can be computed using any oracle satisfying an escape property, which includes the class of non-generalised low₂ sets. Finally, we show that the analogue of Rosenstein's theorems do not hold for the property of not computably embedding a copy of ζ , the order type of the integers, by constructing a computable partial ordering which does not embed ζ , but such that every Δ_2^0 linear extension of the ordering does admit a computable embedding of ζ .

1. DEDICATION

This paper is dedicated to the memory of S. Barry Cooper. Barry was a valued teacher, colleague and friend to us all. We all completed our PhD theses under Barry's supervision, and our subsequent careers bear the hallmarks of his influence. We each benefited greatly from his enthusiasm and generosity. Barry was a purveyor of ideas who, despite the enormous demands on his time and energy made by his various roles within and outside the university, remained generous with his time and advice. This paper reflects that. The main questions considered in this paper were originally articulated by Barry. It was his insight and perseverance that made possible the joint work presented below. When Barry died on the 26th October 2015 we lost both a mentor and a friend. We lost a man who unreservedly shared with each of us his vision of mathematics and the wider world. He is greatly missed.

— James, Charles, Kyung Il and Anthony

2. INTRODUCTION

A common question in the study of partial orders is: which properties Q of a partial order \mathcal{P} can be preserved when taking a linear extension of \mathcal{P} ? That is to say,

$$\begin{aligned} &\text{if } \mathcal{P} \text{ is a partial order satisfying property } Q, \\ &\text{is there a linear extension } \mathcal{L} \text{ of } \mathcal{P} \text{ which also satisfies } Q? \end{aligned} \quad (2.1)$$

Prominent examples of properties Q for which this is true include the property of being well founded, due to Bonnet [Bon69], and the property of being scattered, due to Bonnet & Pouzet [BP82] and Galvin & MacKenzie (unpublished). ‘Well founded’ and ‘scattered’ mean, respectively, not embedding the order type ω^* of the negative integers, and the order type η of the rational numbers. (Precise definitions of these and other notions referred to above are deferred until section 3.)

From a computability perspective, we may ask:

$$\begin{aligned} &\text{if } \mathcal{P} \text{ is a } \textit{computable} \text{ partial order satisfying property } Q, \\ &\text{is there a } \textit{computable} \text{ linear extension } \mathcal{L} \text{ of } \mathcal{P} \text{ which also satisfies } Q? \end{aligned} \quad (2.2)$$

In the case where Q is the property of being well founded, Rosenstein and Kierstead (in [Ros84]) showed that the answer is yes: every well founded computable partial order does have a well founded computable linear extension. For the property of being scattered, the answer was later shown to be ‘no’ by Downey, Hirschfeldt, Lempp and Solomon [DHLS03].

Rosenstein and Kierstead’s result is a partial effectivisation of Bonnet’s theorem that every well founded partial order has a well founded linear extension. However, Rosenstein noted that it is not a complete effectivisation as, although the notion of order has been effectivised (by the introduction of the adjective *computable*), the notion of well foundedness has not. Therefore, Rosenstein proposed a weaker form of the properties of well foundedness and scatteredness. An order \mathcal{P} is *computably well founded* (respectively, *computably scattered*) if there is no *computable* embedding of ω^* (respectively, η) into \mathcal{P} . Rosenstein and Statman (in [Ros84]) showed that (2.2) does not hold for the property of computable well foundedness: there is a computably well founded, computable partial order \mathcal{P} which does not have a computably well founded, computable linear extension. Their counterexample \mathcal{P} was a computable tree with no computable infinite path.

If the answer to (2.2) is ‘no’ for a particular property Q , then we may ask the more general question, what is the minimum complexity of a linear extension \mathcal{L} of \mathcal{P} needed to preserve property Q ? The complexity of an order may be measured, for example, by its position in the arithmetical hierarchy, or by the type of oracle needed to compute the order. Rosenstein took the first steps toward answering this question for the properties of computable well foundedness and computable scatteredness, offering the following two theorems in [Ros84].

Theorem 2.1 (Rosenstein). *Every computably well founded computable partial order \mathcal{P} has a computably well founded linear extension \mathcal{L} which is Δ_2^0 .*

Theorem 2.2 (Rosenstein). *Every computably scattered computable partial order \mathcal{P} has a computably scattered linear extension \mathcal{L} which is Δ_2^0 .*

The proof of Theorem 2.1 uses an oracle for the halting set \emptyset' to construct the required linear extension (thus yielding a Δ_2^0 order by the Limit Lemma, given below as Lemma

3.1). No proof of Theorem 2.2 was given in [Ros84], and to our knowledge a proof of this theorem has not yet appeared in publication.

Rosenstein's theorems thus give an upper bound of Δ_2^0 for the minimum complexity of a linear extension required to preserve the properties of computable well foundedness and computable scatteredness. In a survey chapter [Dow98] for the Handbook of Recursive Mathematics, Downey asks whether this bound on the minimum complexity can be reduced any further. Rosenstein's theorems give the best possible bound within the arithmetical hierarchy, since any Π_1^0 or Σ_1^0 linear order with computable domain is in fact computable.

Therefore, to further sharpen our understanding of the minimum complexity, two approaches present themselves: to investigate the complexity within a more fine-grained subhierarchy of the class of Δ_2^0 sets, or to use another measure of complexity aside from the arithmetical hierarchy.

In this paper, taking the first approach, we investigate the problem using the Ershov hierarchy, a well-known subhierarchy of the class of Δ_2^0 sets. In section 4, we show that the bound on the minimum complexity can be reduced to ω -c.e. within the Ershov hierarchy. That is, every computably well founded (respectively, computably scattered) computable partial order has a computably well founded (computably scattered) linear extension which is ω -c.e. In section 5, we show that this bound cannot be reduced any further within the Ershov hierarchy, by constructing a computably well founded (respectively, classically scattered) computable partial order \mathcal{P} which does not have a computably well founded (computably scattered) linear extension \mathcal{L} which is n -c.e. for any $n < \omega$. We thus identify the minimum level in the Ershov hierarchy at which linear extensions preserving the properties of computable well foundedness and computable scatteredness may be found.

As a step towards the second approach, in section 4 we also show that any oracle satisfying an escape property, which includes all non-generalised low₂ (non-GL₂) sets, can compute a computably well founded (respectively, computably scattered) linearisation of a computably well founded (computably scattered) computable partial order. Thus the information content of the oracle \emptyset' used in Rosenstein's proof of Theorem 2.1 can be reduced. To conclude, we turn our attention briefly to the property of not computably embedding the order type ζ of the integers. We show that in this case the minimum complexity required to preserve the property of not computably embedding ζ is higher: we construct a computable linear order \mathcal{P} which does not embed ζ , but such that every Δ_2^0 linearisation of \mathcal{P} admits a computable embedding of ζ .

3. PRELIMINARIES

We assume $\{W_e\}_{e \in \mathbb{N}}$ to be a listing of c.e. sets with associated c.e. approximation $\{W_e[s]\}_{e,s \in \mathbb{N}}$, such that $x \in W_e[s] \implies x < s$ and $|W_e[s+1] \setminus W_e[s]| \leq 1$ for all e and s . We use \emptyset' to denote the halting set $\{e \mid e \in W_e\}$ and $A \leq_T B$ to signify that the set A is *Turing reducible* to set B (or is *B-computable*)—meaning that there is some Turing machine with oracle B that computes A (under the identification of a set with its characteristic function). $\langle \cdot, \cdot \rangle$ is a standard 1-1 computable pairing function over the integers satisfying $\max\{n, m\} \leq \langle n, m \rangle$ for all $n, m \in \mathbb{N}$.

Lemma 3.1 (Schoenfield). *Let A be a set. The following are equivalent.*

- 1) $A \leq_T \emptyset'$.

- 2) A is Δ_2^0 .
- 3) There is a computable function $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ such that
 - (a) For all $n \in \mathbb{N}$, $f(n, 0) = 0$.
 - (b) For all $n \in \omega$, $\lim_{s \rightarrow \infty} f(n, s) = A(n)$.

We generally use notation of the form $A(n)[s]$ to denote such $f(n, s)$ and $\{A[s]\}_{s \in \mathbb{N}}$ to denote the derived approximation to A —where we may also think of $A[s]$ as a (finite) set. We say that such an approximation is Δ_2^0 .

Ershov's Hierarchy [Ers68a, Ers68b, Ers70, Ars11] gives a classification of the Δ_2^0 sets in terms of (notations of) the computable ordinals. Roughly speaking the place of a Δ_2^0 set A in the Ershov Hierarchy is given by measuring how long it takes for an optimal (in terms of time) Δ_2^0 approximation $\{A[s]\}_{s \in \mathbb{N}}$ to A to settle down on any input $n \in \mathbb{N}$ —where we say that $\{A[s]\}_{s \in \mathbb{N}}$ has *settled down* at stage t on input n if $A(n)[s] = A(n)$ for all $s \geq t$.

The present paper is concerned with the initial segment of the Ershov Hierarchy defined relative to the finite ordinals $0, 1, 2, \dots$ and ω . Accordingly we apply the following commonly used variant of Ershov's original definition.

Definition 3.2. A Δ_2^0 set A is said to be ω -*computably enumerable* (or ω -*c.e.*) if there is a Δ_2^0 approximation $\{A[s]\}_{s \in \mathbb{N}}$ to A and computable function $g : \mathbb{N} \rightarrow \mathbb{N}$ such that

$$|\{s \mid A(n)[s+1] \neq A(n)[s]\}| \leq g(n)$$

for all $n \in \mathbb{N}$. A is said to be k -*computably enumerable* (or k -*c.e.*) for finite k if we can choose $g : \mathbb{N} \rightarrow \{k\}$, i.e. if $A(n)[s+1] \neq A(n)[s]$ for at most k stages s . For $\gamma \leq \omega$, we use Σ_γ^{-1} to denote the class of γ -c.e. sets and $\Sigma_{<\omega}^{-1}$ to denote $\bigcup_{k < \omega} \Sigma_k^{-1}$, i.e. the class of all sets that are k -c.e. for some finite k .

We note the existence of a listing $\{R_{\langle k, e \rangle}\}_{k, e \in \mathbb{N}}$ —which we shall use in section 5—of the class $\Sigma_{<\omega}^{-1}$ with computable approximation $\{R_{\langle k, e \rangle}[s]\}_{k, e, s \in \mathbb{N}}$ where, for any $k, e \in \mathbb{N}$, $\{R_{\langle k, e \rangle}[s]\}_{s \in \mathbb{N}}$ is a $k+1$ -c.e. approximation to the $k+1$ -c.e. set $R_{\langle k, e \rangle}$. Indeed this easily follows¹ from the result (originally due to Ershov) that such a listing exists for any fixed $k+1$, by observing that the associated ($k+1$ -c.e.) approximation is constructed uniformly in $k+1$.

¹Another way of doing this directly is in terms of the representation of n -c.e. sets as the union of differences of c.e. sets. Accordingly—letting $(e)_i$ denote the i th projection of e coded as a sequence of length $n+1$ —define $g : \mathbb{N}^3 \rightarrow \mathbb{N}$ such that

$$W_{g(n, e, k)} = \begin{cases} \bigcap_{0 \leq i \leq k} W_{(e)_i} & \text{if } 0 < k \leq n \\ \emptyset & \text{otherwise.} \end{cases}$$

Note that n dictates the choice of sequence into which we decode e (using the coding of finite sequences induced by our pairing function). Thus if $n = 2$ the code $e = \langle d_0, d_1, d_2 \rangle$ is used with $(e)_i = d_i$. Notice that, by convention, if $n = 0$, $e = \langle e \rangle$ so that $(e)_0 = e$. We now set

$$R_{\langle n, e \rangle} = \bigcup_{i=0}^{\lfloor \frac{n}{2} \rfloor} W_{g(n, e, 2i)} \setminus W_{g(n, e, 2i+1)}$$

with $R_{\langle n, e \rangle}[s]$ defined by using the s stage approximations of the c.e. sets involved in its definition. It is easy to check that $\{R_{\langle n, e \rangle}[s]\}_{s \in \mathbb{N}}$ defines an $n+1$ -c.e. approximation to $R_{\langle n, e \rangle}$. For example, if $e = \langle e \rangle = \langle c_0, c_1 \rangle = \langle d_0, d_1, d_2 \rangle$ then $R_{\langle 0, e \rangle} = W_e$, $R_{\langle 1, e \rangle} = W_{c_0} \setminus (W_{c_0} \cap W_{c_1})$ and

$$R_{\langle 2, e \rangle} = W_{d_0} \setminus (W_{d_0} \cap W_{d_1}) \cup (W_{d_0} \cap W_{d_1} \cap W_{d_2}),$$

We say that $\mathcal{P} = (P, <_P)$ is a *partial order* if $<_P$ is irreflexive, asymmetric and transitive. For (distinct) $a, b \in P$ if neither $a <_P b$ nor $b <_P a$ we say that a, b are *$<_P$ -incomparable* written $a \upharpoonright_P b$ and we say that a, b are *$<_P$ -comparable* otherwise. We use $(a, b)_P$ to denote the subpartial order of \mathcal{P} made up of those element lying strictly $<_P$ -between a and b and we use square brackets to indicate inclusion of the limiting elements. (E.g. $[a, b]_P$ includes both a and b .) We refer to the latter as *$<_P$ -intervals*. Note that in Section 5, having duly forewarned the reader, we drop the “ P ” subscript in our notation for intervals. We use the notation $(a, b)_{P^*}$ to denote the *set* of elements lying strictly $<_P$ -between a and b in \mathcal{P} —where the actual $<_P$ -ordering of a and b is not significant. For convenience we use $-\infty$ and $+\infty$ to denote notional elements such that $-\infty <_P x$ and $x <_P +\infty$ for all x . We say that $\mathcal{L} = (L, <_L)$ is a *linear order* (that we sometimes also call a *chain*) if, for all distinct $a, b \in L$, a and b are $<_L$ -comparable. (I.e. $a <_L b$ or $b <_L a$.) We say that \mathcal{L} is a *linear extension of* (or *linearises*) \mathcal{P} if $L = P$ and, for all $a, b \in P$ $a <_P b \Rightarrow a <_L b$.

For a computational complexity class Γ (such as $\Delta_1^0, \Sigma_1^0, \Pi_1^0, \Sigma_{<\omega}^{-1}, \Sigma_{\omega}^{-1}$ or Δ_2^0) we say that $\mathcal{P} = (P, <_P)$ is Γ if both P and $<_P$ are Γ . If the domain P is in fact computable we usually make the identification $\mathcal{P} = (\mathbb{N}, <_P)$ via a (computably invertible) computable labelling of P . Note that any linear extension of a computable partial order \mathcal{P} has computable domain by definition.

We think of $<_P$ as both a subset of $\mathbb{N} \times \mathbb{N}$ and a characteristic function over \mathbb{N} under our pairing function $\langle \cdot, \cdot \rangle$. Accordingly $<_P(\langle n, m \rangle) = 1$ if and only if $n <_P m$. We also generalise this notation to that of approximations of sets with $<_P(\langle n, m \rangle)[s]$ denoting the s -stage approximation to $<_P(\langle n, m \rangle)$. We sometimes use letters such as R to denote the order relation, for example using (\mathbb{N}, R) to denote a partial/linear order.

A linear order type α is said to be *computable* if there exists a computable linear order $\mathcal{L} = (L, <_L)$ of type α . Note that we also refer to \mathcal{L} as a *computable copy* of α in this case. We say that partial order $\mathcal{P} = (P, <_P)$ (*computably*) *embeds* order type α if there is a (computable) copy $\mathcal{L} = (L, <_L)$ of α such that $L \subseteq P$ and $<_L$ coincides with $<_P$ over L —i.e. such that $\mathcal{L} = (L, <_P \upharpoonright L)$. For two order types α, β we say that β embeds α —written $\alpha \leq \beta$ —if there is a copy \mathcal{L}_β of β which embeds α . $1 + \alpha$ ($\alpha + 1$) denotes α with a bottom (top) element adjoined. More generally $\alpha + \beta$ denotes the linear sum of α and β . ω, ω^*, ζ and η denote respectively the order types of the nonnegative integers, negative integers, integers, and rationals.

A partial order is said to be (*computably*) *well founded* if it does not (computably) embed the order type ω^* . Note that it is easily shown that \mathcal{P} is well founded if and only if there is no infinite descending sequence $x_0 >_P x_1 >_P x_2 >_P \dots$ contained in \mathcal{P} —which we refer to as an ω^* sequence—and also that \mathcal{P} is computably well founded if and only if, for all indices e , $\{W_e[s]\}_{s \in \mathbb{N}}$ does not enumerate such a sequence. We say that \mathcal{P} is (*computably*) *scattered* if \mathcal{P} does not (computably) embed the order type of the rationals η . Now, defining the *dyadic* function $d : \mathbb{N} \rightarrow \mathbb{Q} \cap (0, 1)$ by setting

$$d(n) = \begin{cases} \frac{1}{2} & \text{if } n = 0 \\ \frac{1+2m}{2^{k+1}} & \text{if } n = (2^k - 1) + m \text{ for some } k \geq 1 \text{ and } 0 \leq m < 2^k, \end{cases}$$

from which it is obvious that the associated approximations (e.g. $R_{(0,e)}[s] = W_e[s]$ etc.) are respectively c.e., 2-c.e. and 3-c.e. Note also that it is also easy to show that $\{R_{(n,e)}\}_{n,e \in \mathbb{N}}$ lists all the $n + 1$ -c.e. sets.

we say that x_0, x_1, x_2, \dots is an η -sequence in \mathcal{P} if, for all distinct $n, m \geq 0$, $x_n <_{\mathcal{P}} x_m \Leftrightarrow d(n) < d(m)$. Note that we can also think of the x_i as labelling nodes on a binary tree—and thus alternatively use binary instead of dyadic representations of the indices of the sequence—so that x_0 labels the root and introduction of the labels of nodes of length $l+1$ corresponds to “densifying” (from left to right) the labels of nodes of length $\leq l$. For example, after enumerating 7 elements of our sequence—i.e. those elements labelled by nodes of length ≤ 2 —we have:

$$x_3 <_{\mathcal{P}} x_1 <_{\mathcal{P}} x_4 <_{\mathcal{P}} x_0 <_{\mathcal{P}} x_5 <_{\mathcal{P}} x_2 <_{\mathcal{P}} x_6.$$

Now it is also easily shown that \mathcal{P} is scattered if and only if there is no η sequence contained in \mathcal{P} and is computably scattered if and only if, for all indices e , $\{W_e[s]\}_{s \in \mathbb{N}}$ does not enumerate such a sequence.

Remark. Using these notions and the methods used below and in [Gay16] we can build, via computable constructions, ω -c.e. copies—with computable domain— \mathcal{L}_{ω^*} and \mathcal{L}_{η} of ω^* and η respectively, such that \mathcal{L}_{ω^*} is computably well founded and \mathcal{L}_{η} is computably scattered.

Further background on the basic computability theoretic techniques used here can be found in [Coo04, Soa87, Odi89, Web12]. We also refer the reader to [Dow98] for a review of computability theoretic results in the context of linear orderings.

4. EXTENDIBILITY RESULTS

In this section we show that, for both of the properties computable scatteredness and computable well foundedness and any computable partial order which satisfies one of these properties, there exists an ω -c.e. linearisation of \mathcal{P} that also satisfies that property.

Remark. Given a computable order type α and a computable copy of η , we know that η computably embeds α . Note in particular that this implies that any partial order \mathcal{P} which is computably well founded is also computably scattered.

Before considering Theorem 4.1 below we note that an alternative proof of this result using the notion of an η -sequence can be found in Chapter 4 of [Gay16].

Theorem 4.1. *Every computably scattered computable partial order has a computably scattered ω -c.e. linear extension.*

Proof. Let $\mathcal{P} = (\mathbb{N}, <_{\mathcal{P}})$ be a computably scattered computable partial order. We will build a linear order $\mathcal{L} = (\mathbb{N}, <_{\mathcal{L}})$ via a computable construction where at each stage s we define a finite approximation $\mathcal{L}[s] = (L[s], <_{\mathcal{L}}[s])$ to \mathcal{L} where $<_{\mathcal{L}}[s]$ is a linearisation of $<_{\mathcal{P}}$ restricted to domain $L[s]$. These approximations are defined such that $L[0] = \emptyset$ and, for all $s > 0$, $L[s] = \{0, \dots, \lfloor \frac{s}{2} \rfloor\}$ (so that $L[2t] = L[2t+1] = \{0, \dots, t\}$). Defining

$$<_{\mathcal{L}}(\langle n, m \rangle) = \lim_{s \rightarrow \infty} <_{\mathcal{L}}(\langle n, m \rangle)[s]$$

the construction will ensure that $<_{\mathcal{L}}(\langle n, m \rangle)$ is defined for all $n, m \in \mathbb{N}$ and moreover that $|\{s \mid <_{\mathcal{L}}(\langle n, m \rangle)[s] \neq <_{\mathcal{L}}(\langle n, m \rangle)[s+1]\}| < 2^{\langle n, m \rangle}$ so that $<_{\mathcal{L}}$ is ω -c.e. Our construction will also be defined to satisfy for all indices e the following requirements:

Q_e : if W_e is infinite then there exist distinct $x, y \in W_e$ such that $W_e \cap (x, y)_{L^*} = \emptyset$.

In this way W_e cannot be the domain of a dense linear order embedded in \mathcal{L} so that satisfaction of Q_e for all e implies that \mathcal{L} is computably scattered². The requirements are ordered in the usual way so that Q_i has higher priority than Q_j if $i < j$.

Notes and Notation.

For any index e we let x_0^e, x_1^e, \dots denote the members of W_e in the order dictated by the approximation $\{W_e[s]\}_{s \in \mathbb{N}}$ of W_e . We say, if x_i^e, x_j^e are both defined, that x_i^e is *e-older than* x_j^e if $i < j$ (and likewise that x_j^e is *e-younger than* x_i^e) and we apply this terminology directly without use of indices.

The strategy to satisfy requirement Q_e is to find a suitable pair $\{x_e, y_e\} \subseteq W_e$ —a *witness*—and to endeavour to ensure that the interval $(x_e, y_e)_L$ remains empty. Formally, a *witness* is a tuple $\langle e, x_e, y_e \rangle$ where $e \in \mathbb{N}$, $x_e, y_e \in W_e$. The construction will ensure that $x_e <_L y_e$ for such witnesses. A *witness set* is a finite set of witnesses M such that there is at most one tuple $\langle e, x_e, y_e \rangle \in M$ for each e . In addition to the ordering $\mathcal{L}[s]$, our construction will define a witness set $M[s]$ at each stage s . Define

$$t_e[s] = \begin{cases} \max\{x_e, y_e\} & \text{if there is some } \langle e, x_e, y_e \rangle \in M[s] \\ 0 & \text{otherwise,} \end{cases}$$

and note that $t_e[s]$ essentially specifies the ‘portion’ of \mathcal{L} that Q_e wishes to preserve in order to ensure $(x_e, y_e)_{L^*} = \emptyset$. We also define

$$T_e[s] = \max \{ t_i[s] \mid i < e \} \cup \{e\} \tag{4.1}$$

which is the (length of the) initial segment of \mathbb{N} on which Q_e is not permitted to modify $<_L$. Accordingly, elements smaller than $T_e[s]$ act as “barriers” against action taken by requirement Q_e at stage $s + 1$ in order to preserve the work of higher priority requirements (with the term $\{e\}$ added to ensure that $<_L$ is ω -c.e.).

We now define some procedures which will be used in the construction as well as in Theorem 4.9. The first procedure takes as input an ordering $\mathcal{J} = (J, <_J)$ on a finite set $J = \mathbb{N} \upharpoonright k$ for some k , and a witness set M of witnesses to be respected, such that \mathcal{J} is a linear extension of $\mathcal{P} \upharpoonright J = (J, <_P \upharpoonright J)$, $x <_J y$ and $(x, y)_J$ is empty for any $\langle i, x, y \rangle \in M$. The procedure extends the ordering \mathcal{J} by one element, by inserting the new element as high as possible in the ordering while respecting the witnesses, i.e. maintaining $(x, y)_J$ empty.

Procedure $P_1(\mathcal{J}, M)$: Single-point extension of \mathcal{J} respecting M .

Let $n = |J|$. Produce a new linear order \mathcal{J}' on domain $J' = J \cup \{n\}$ (i.e., extending the domain of J by one) by inserting n as follows.

Case 1. There is no $m \in J$ with $n <_P m$. Then insert n as the top element in \mathcal{J}' , thus defining $x <_{J'} n$ for all $x \in J$.

Case 2. There is some $m \in J$ with $n <_P m$. Let m be the $<_J$ -least such. There are two subcases.

²It follows from the Remark at the end of the Section 3 that the class \mathcal{C} of computable copies of η is a proper subclass of the class \mathcal{D} of copies of η whose domain is computable. Therefore our proof rules out any embedding in \mathcal{L} of a member of \mathcal{D} and so, in effect, delivers more than just computable scatteredness of \mathcal{L} .

Case 2a. m is not equal to y_i for any $\langle i, x_i, y_i \rangle \in M$. Then obtain \mathcal{J}' by inserting n as the immediate predecessor of m in \mathcal{J} , thus defining $x <_{\mathcal{J}'} n$ for $x \in J$ with $x <_J m$, and $n <_{\mathcal{J}'} x$ for $x \in J$ with $m <_J x$ or $x = m$.

Case 2b. m is equal to y_i for some $\langle i, x_i, y_i \rangle \in M$. Then obtain \mathcal{J}' by inserting n as the immediate predecessor of x_i in \mathcal{J} , thus defining $x <_{\mathcal{J}'} n$ for $x \in J$ with $x <_J x_i$, and $n <_{\mathcal{J}'} x$ for $x \in J$ with $x_i <_J x$ or $x = x_i$.

The result of $P_1(\mathcal{J}, M)$ is the order $\mathcal{J}' = (J', <_{\mathcal{J}'})$ obtained above.

End of Procedure P_1 .

The next procedure creates a new empty interval $(x_e, y_e)_{\mathcal{J}'}$ to satisfy Q_e , via a method we call *plummeting*. The procedure takes as input a linear order $\mathcal{J} = (J, <_J)$, a pair $p = \{x, y\} \subseteq J$ such that $x|_P y$ and an index e of a c.e. set such that $x, y \in W_e$.

Procedure $P_2(\mathcal{J}, p, e)$: e -plummeting the pair $p = \{x, y\}$ in the order \mathcal{J} .

Assume without loss of generality that $x <_J y$. Let S be the $<_J$ -ordered set lying in the $<_J$ -interval $(x, y)_J$ comprising precisely those elements z such that $x <_P z$. If y is e -older than x , then we remove the set $\{x\} \cup S$ and reinsert it immediately above y to obtain \mathcal{J}' , whereas if y is e -younger than x , then we reinsert S immediately above y and x immediately below y . We do this in such a way that the $<_J$ -order of $\{x\} \cup S$ is preserved. For example, if $S = \{x_0 <_J x_1 <_J \cdots <_J x_k\}$ and y is e -older than x , then $y <_{\mathcal{J}'} x$ and the $<_{\mathcal{J}'}$ -interval $[y, x_k]_{\mathcal{J}'}$ in \mathcal{J}' is precisely the chain

$$y <_{\mathcal{J}'} x <_{\mathcal{J}'} x_0 <_{\mathcal{J}'} x_1 <_{\mathcal{J}'} \cdots <_{\mathcal{J}'} x_k \quad (4.2)$$

whereas if y is e -younger than x , then $x <_{\mathcal{J}'} y$ and the $<_{\mathcal{J}'}$ -interval $[x, x_k]_{\mathcal{J}'}$ is the chain in (4.2) with the order of x and y reversed. Note that x and y are $<_{\mathcal{J}'}$ -juxtaposed (i.e. adjacent) in both cases.

The result of $P_2(\mathcal{J}, p, e)$ is the order $\mathcal{J}' = (J, <_{\mathcal{J}'})$ where $<_{\mathcal{J}'}$ is as above.

End of Procedure P_2 .

Note that, in the present construction the use of the ages of x and y is irrelevant beyond the fact that it helps to fully determine the plummeting process. However, as we shall see, it allows the present construction to be applied directly in Theorem 4.8.

At each stage s of the construction we will define a finite linear order $\mathcal{L}[s]$ and a set of witnesses $M[s] = \{\langle i, x_i, y_i \rangle : i \in D\}$ where D is a finite subset of \mathbb{N} . If there is some $\langle i, x_i, y_i \rangle \in M[s]$ then we say that Q_i is *on* at the end of stage s ; otherwise we say that Q_i is *off* at the end of stage s .

Q_e *requires attention* at stage $s + 1$ if it is *off* at the end of stage s and there are distinct numbers x, y such that

- (i) $x, y \in L[s] \cap W_e[s]$, and
- (ii) $x|_P y$, and
- (iii) both x, y and every number lying $<_L[s]$ between them is greater (under $<_{\mathbb{N}}$) than $T_e[s]$.

The Construction

Stage 0. Define \mathcal{L} to be the trivial order on $L[0] = \emptyset$ and define $M[0] = \emptyset$.

Stage $s + 1$ with s even. We add $n = \frac{s}{2}$ to the order.

Let $\mathcal{L}[s+1]$ be the order obtained by running Procedure $P_1(\mathcal{L}[s], M[s])$.³

Declare $M[s+1] = M[s]$ and proceed to stage $s+2$.

Stage $s+1$ with s odd. We look to take action for some requirement Q_e by e -plummeting.

Case 1. Some requirement Q_e with $e < \frac{s+1}{2}$ requires attention.⁴

In this case choose the least such e and the least (under coding $\langle \cdot, \cdot \rangle$) pair x, y satisfying (i)-(iii). Let $\mathcal{L}[s+1]$ be the order obtained by running Procedure $P_2(\mathcal{L}[s], \{x, y\}, e)$. Let x_e be the e -older of the pair $\{x, y\}$ and y_e be the e -younger. Let $M[s+1] = \{\langle i, x_i, y_i \rangle : \langle i, x_i, y_i \rangle \in M[s] \text{ and } i < e\} \cup \{\langle e, x_e, y_e \rangle\}$ (hence Q_e becomes *on* and all Q_j with $j > e$ become *off*). We say that Q_e receives attention in this case.

Case 2. Otherwise.

Simply define $\mathcal{L}[s+1] = \mathcal{L}[s]$ and $M[s+1] = M[s]$.

Verification.

Let $\text{WHA}[s]$ denote our *Working Hypothesis A* at stage s defined to be the conjunction of Conditions (1)-(3) below.

- (1) $\mathcal{L}[s]$ is a linear extension of $\langle_P \upharpoonright L[s]$.
- (2) If Q_e is *on* at the end of stage s then there is a (unique) pair x_e, y_e such that $\langle e, x_e, y_e \rangle \in M[s]$ and the $\langle_L[s]$ -interval $(x_e, y_e)_{L[s]}$ is empty. Also x_e is e -older than⁵ y_e . (And if Q_e is *off* then there are no such elements.)
- (3) For any $\langle i, x, y \rangle$ and $\langle j, x', y' \rangle \in M[s]$ with $i \neq j$, $\{x, y\} \cap \{x', y'\} = \emptyset$.

Remark. Condition 3 of $\text{WHA}[s]$, when taken in conjunction with Condition 2, says that, for witnesses $\langle i, x, y \rangle$ and $\langle j, x', y' \rangle$ at stage s , either $x <_L y <_L x' <_L y'$ or $x' <_L y' <_L x <_L y$. So $\text{WHA}[s]$ implies that there is a linear ordering over the set of witnesses at stage s . Note that the truth of $\text{WHA}[s]$ validates the ‘pre-condition’ of Procedure 1 that $\mathcal{J} = \mathcal{L}[s]$ is a linear extension of $\mathcal{P} \upharpoonright J$, $x <_J y$ and $(x, y)_J$ is empty for $\langle i, x, y \rangle \in M = M[s]$.

A straightforward analysis of the construction and Procedures P_1 and P_2 yields the following.

Lemma 4.2. *For any stage $s \geq 0$ of the construction, $\text{WHA}[s] \Rightarrow \text{WHA}[s+1]$.*

Now also let $\text{WHB}[s]$ denote our *Working Hypothesis B* at stage s defined to be the conjunction of Conditions 4-5 below.

- (4) For any i such that Q_i is *on* at the end of both stage s and $s+1$, and any $x, y \in L[s]$ such that $z \leq t_i[s]$ ($= t_i[s+1]$) for some $z \in \{x, y\}$, $x <_{L[s+1]} y \Leftrightarrow x <_{L[s]} y$.
- (5) For any $x, y \in L[s]$ if, for all $i \leq \max\{x, y\}$, Q_i does not receive attention at stage $s+1$, then $x <_{L[s+1]} y \Leftrightarrow x <_{L[s]} y$.

³Procedure P_1 requires that $x <_J y$ and $(x, y)_J$ is empty for each $\langle i, x, y \rangle \in M$; we verify later that this holds.

⁴The condition $e < \frac{s+1}{2}$ ensures that it is always the case that $\mathbb{N} \upharpoonright (T_e[s] + 1) \subseteq L[s]$ when Q_e requires attention.

⁵This part of the Working Hypothesis is only relevant to the proof of Theorem 4.8.

Define $\text{WH}[s]$ to be our overall *Working Hypothesis* at stage s , i.e. the conjunction $\text{WHA}[s]$ & $\text{WHB}[s]$ of all five conditions. Then our next Lemma follows by induction over stages for $\text{WHA}[s]$ and direct inspection for $\text{WHB}[s]$.

Lemma 4.3. *For any stage $s \geq 0$ of the construction, $\text{WH}[s]$ is true.*

Note 4.4. Suppose that s^* is such that Q_e is *on* at the end of every stage $s \geq s^*$. Note that $T_e[s] = T_e[s^*]$ for all $s \geq s^*$ in this case. Then it follows from Lemma 4.3 that, for any $n, m \in \mathbb{N}$ such that $z \leq T_e[s^*]$ for some $z \in \{n, m\}$, and all $s \geq s_{n,m}$, $n <_L[s] m \Leftrightarrow n <_L[s_{n,m}] m$, where $s_{n,m}$ is the least stage $\geq s^*$ such that $n, m \in L[s_{n,m}]$. In other words $<_L(\langle n, m \rangle)$ is defined and coincides with $<_L(\langle n, m \rangle)[s]$ for all $s \geq s_{n,m}$. Moreover Lemma 4.3 also implies that Q_e will be satisfied in this case as the witness $\langle e, x_e, y_e \rangle \in M[s^*]$ remains in $M[s]$ henceforth and the $<_L$ -interval $(x_e, y_e)_L$ remains empty.

Lemma 4.5. *For all $e \geq 0$, Q_e receives attention at most 2^e times and is satisfied⁶.*

Proof. Consider index $e > 0$ and suppose that there is a stage s_e such that, for all $s > s_e$ and $i < e$, Q_i does not receive attention at stage s . Suppose also that s_e is the least such stage. As already mentioned in Note 4.4 we know that $T_e[s] = T_e[s_e]$ for all $s \geq s_e$ so we can define $T_e = \lim_{s \rightarrow \infty} T_e[s] = T_e[s_e]$. Label $\mathbb{N} \upharpoonright (T_e + 1)$ as $\{b_0, \dots, b_{T_e}\}$ in such a way that

$$b_0 <_L[s_e] b_1 <_L[s_e] \dots <_L[s_e] b_{T_e}$$

and notice that, by Note 4.4, this ordering coincides with $<_L[s]$ for all $s \geq s_e$ and so also with $<_L$. (I.e. the ordering of the b_i 's is terminally fixed from stage s_e onwards.) Letting b_{-1} and b_{T_e+1} denote $-\infty$ and $+\infty$ respectively, for $0 \leq i \leq T_e + 1$ define the sets

$$U_i = W_e \cap \{z \in \mathbb{N} \mid b_{i-1} <_L z <_L b_i\}$$

(i.e. $U_i = W_e \cap (b_{i-1}, b_i)_{L^*}$) and observe that, by Note 4.4, U_i is a c.e. set with c.e. approximation $\{U_i[s]\}_{s \in \mathbb{N}}$ defined by setting

$$U_i[s] = \begin{cases} \emptyset & \text{if } s < s_e \\ W_e[s] \cap L[s] \cap \{z \in \mathbb{N} \mid b_{i-1} <_L z <_L b_i\} & \text{otherwise.} \end{cases}$$

Remark. Intuitively Note 4.4 tells us that, from stage s_e onwards, the b_i 's act as a set of fixed barriers across which nothing in the construction now moves. In other words any reordering of elements in the ongoing approximation $\mathcal{L}[s]$ either occurs strictly below b_0 or strictly above b_{T_e} or strictly in between $b_{i-1} <_L b_i$, for some $1 \leq i \leq T_e$ and so, by definition, this involves two elements x, y that are both in the same set U_j .

Suppose that W_e is infinite. Then we can choose i such that U_i is infinite. If there exist distinct elements $x, y \in U_i$ such that $x \mid_P y$ then clearly Q_e will receive attention at some stage $s > s_e$ and be satisfied by Note 4.4. Otherwise U_i is linearly ordered by $<_P$. This means that $<_L$ is defined and coincides with $<_P$ over U_i . However computable scatteredness of \mathcal{P} implies that there is some pair $x, y \in U_i$ such that $U_i \cap (x, y)_{L^*}$ is empty. So, as the latter is the same as $W_e \cap (x, y)_{L^*}$, Q_e is once again satisfied.

Now suppose that Q_i in fact receives attention at most 2^i times for all $i \leq e$ so that this condition applies to all such i at stage s_e . By inspection—using the observation

⁶Note that we are not assuming that $<_L(\langle n, m \rangle)$ is defined for all $n, m \in \mathbb{N}$.

that for as long as a requirement is *on* it does not receive attention—we see that this implies that Q_e has received attention at most $(2^{e-1} - 1) \times 2 + 1 = 2^e - 1$ times by (the end of) stage s_e . But, noting in the above that Q_e receives attention at most once after stage s_e , we see that Q_e receives attention at most 2^e times.

As the argument for $e = 0$ is an easy variant of the argument applied to the case $e > 0$, Lemma 4.5 follows by induction over indices $e \geq 0$. \square

Lemma 4.6. *For all $n, m \in \mathbb{N}$, $<_L(\langle n, m \rangle)$ is defined and*

$$|\{s \mid <_L(\langle n, m \rangle)[s+1] \neq <_L(\langle n, m \rangle)[s]\}| < 2^{\langle n, m \rangle}.$$

In other words $<_L$ is ω -c.e.

Proof. Note firstly that, for any $n, m \in \mathbb{N}$, $\max\{n, m\} \leq \langle n, m \rangle$. Also (by definition of T_e), for any stage $s + 1$, a requirement Q_i can only change the $<_L$ -order of n and m at stage $s + 1$ if $i < \max\{n, m\}$.⁷ It follows by Lemma 4.5 that there exist at most $\sum_{i=0}^{\langle n, m \rangle - 1} 2^i = 2^{\langle n, m \rangle} - 1$ such stages $s + 1$. \square

Lemma 4.7. $\mathcal{L} = (\mathbb{N}, <_L)$ is a linear extension of \mathcal{P} .

This Lemma follows directly from the conjunction of Lemmas 4.3 and 4.6 and concludes the proof of Theorem 4.1. \square

We now reapply the above construction to prove Theorem 4.8 below. We note that this result was first proved in Chapter 2 of [Lee11].

Theorem 4.8. *Every computably well founded computable partial order has a computably well founded ω -c.e. linear extension.*

Proof. Let $\mathcal{P} = (\mathbb{N}, <_P)$ be a computably well founded computable partial order and construct the linear order $\mathcal{L} = (\mathbb{N}, <_L)$ precisely as in the proof of Theorem 4.1. In this case we need to satisfy, for all indices e , the following requirement:

$$Q_e : \text{if } W_e \text{ is infinite then there exist } x, y \in W_e \text{ such that } x \text{ is } e\text{-older than } y \\ \text{and } x <_L y.$$

This means that $\{W_e[s]\}_{s \in \mathbb{N}}$ does not enumerate an ω^* sequence in \mathcal{L} so that satisfaction of Q_e for all e implies that \mathcal{L} is computably well founded.

Apply the same Verification as that in the proof of Theorem 4.1 except for the analysis of $<_L$ relative to the sets U_i in Lemma 4.5. Here, under the assumption that W_e is infinite we consider the least i such that U_i is nonempty. If U_i is finite then there is some $j > i$ and $x \in U_i, y \in U_j$ such that x is e -older than y . But, by Note 4.4 (and the definitions of the elements b_k and sets U_k), we know that, $x <_L b_i, b_{j-1} <_L y$ and that if $i < j - 1$ then $b_i <_L b_{j-1}$, so that $x <_L y$. On the other hand, if U_i is infinite, and there exist distinct $x', y' \in U_i$ such that $x' \mid_P y'$ then clearly Q_e will receive attention at some stage $s > s^*$ and will thus be satisfied, as Procedure P_2 ensures that the e -oldest of x', y' is placed $<_L$ -below the e -youngest. Otherwise U_i is (infinite and) linearly ordered by $<_P$ and so $<_L$ coincides with $<_P$ over U_i . However, as U_i is itself a c.e. set, computable well foundedness of \mathcal{P} implies that there are elements $x, y \in U_i$ such that x is e -older than y and $x <_L y$. Thus in each case Q_e is satisfied.

⁷In fact we could use $\min\{n, m\}$ instead of $\max\{n, m\}$ here.

Since the rest of the Verification is unchanged we see that not only Lemma 4.5 but also Lemmas 4.6 and 4.7 follow as before so that \mathcal{L} is a ω -c.e. computably well founded linear extension of \mathcal{P} in this case. \square

We now give an extension of Rosenstein's theorems in a different direction. By the Limit Lemma 3.1, Rosenstein's theorems 2.1 and 2.2 show that every computably well founded (respectively, computably scattered) computable partial order \mathcal{P} has a computably well founded (computably scattered) linearisation which is computable in \emptyset' . Indeed, Rosenstein's original proof of Theorem 2.1 uses an oracle for \emptyset' to construct the desired extension. We now show that the oracle \emptyset' can be replaced with any oracle satisfying the following property. Say that $X \subseteq \mathbb{N}$ has the \emptyset' *escape property* if⁸

$$\forall \text{ total functions } f \leq_T \emptyset' \exists \text{ total function } g \leq_T X \text{ s.t. } \exists^\infty x \ g(x) \geq f(x). \quad (4.3)$$

The class of sets satisfying (4.3) includes the non-generalised low₂ (non-GL₂) sets.⁹ Indeed it is well known (see for example [Nie09] Exercise 1.5.21) that X is non-GL₂ if and only if

$$\forall \text{ total functions } f \leq_T \emptyset' \oplus X \exists \text{ total function } g \leq_T X \text{ s.t. } \exists^\infty x \ g(x) \geq f(x)$$

which immediately implies (4.3).

Theorem 4.9. *Let $X \subseteq \mathbb{N}$ satisfy (4.3). For any computably scattered computable partial order \mathcal{P} , there is a computably scattered linear extension \mathcal{L} of \mathcal{P} which is computable in X .*

Proof. We first prove the theorem for the case when $X = \emptyset'$. (This is in essence a proof of Rosenstein's Theorem 2.2.) We later show how to adapt the construction for an oracle satisfying (4.3).

We will build the required linear extension \mathcal{L} in finite extensions. Much of the terminology and notation will be the same as in the proof of Theorem 4.1. We will satisfy the same requirements

$$Q_e : \text{ if } W_e \text{ is infinite then there exist distinct } x, y \in W_e \text{ such that } W_e \cap (x, y)_{L^*} = \emptyset.$$

The basic strategy to satisfy a requirement Q_e at stage $s + 1$ is: repeatedly extend the order $\mathcal{L}[s]$ element-by-element using Procedure P_1 , until we find a suitable pair x, y to plummet using Procedure P_2 in order to satisfy Q_e . Of course, this may not terminate as we may never find a suitable x, y , but that can be determined with the oracle \emptyset' . If the oracle tells us there is no such $\{x, y\}$ then we can argue, similarly to Lemma 4.5, that Q_e is automatically satisfied due to computable scatteredness of $<_{\mathcal{P}}$.

At stage $s + 1$, we will have a linear order $\mathcal{L}[s]$ on some initial segment $L[s]$ of \mathbb{N} , along with a finite witness set $M[s] = \{\langle i, x_i, y_i \rangle : i \in D\}$, and we define $\mathcal{L}[s + 1]$ and $M[s + 1]$. We define one further procedure which will be used in the construction, which takes as input a finite linear order \mathcal{J} , a witness set $M = \{\langle i, x_i, y_i \rangle : i \in D\}$ with $x_i <_{\mathcal{J}} y_i$ and $(x_i, y_i)_{\mathcal{J}}$ empty, and an index e of a c.e. set.

Procedure $P_3(\mathcal{J}, M, e)$: *Search for a candidate to e -plummet.*

Let $T = |J|$ where $\mathcal{J} = (J, <_{\mathcal{J}})$. Build an order $\mathcal{I} = (I, <_{\mathcal{I}})$ in substages.

Substage $t = 0$: Set $\mathcal{I}[0] = \mathcal{J}$.

⁸Recall that $\exists^\infty x P(x)$ means $\forall n \exists x > n P(x)$.

⁹Recall that a set X is generalised low₂ (GL₂) if $X'' \equiv_T (\emptyset' \oplus X)'$.

Substage $t + 1$: Check if there are distinct numbers x, y such that

- (i) $x, y \in W_e[t] \cap I[t]$, and
- (ii) $x|_P y$, and
- (iii) both x, y and every number lying $<_I[t]$ between them is greater (under $<_{\mathbb{N}}$) than T .

If there is such a pair x, y then choose the least such (under $\langle \cdot, \cdot \rangle$) and let \mathcal{I} be the ordering obtained by running Procedure $P_2(\mathcal{I}[t], \{x, y\}, e)$. Say that $P_3(\mathcal{J}, M, e)$ terminates after $t + 1$ steps and returns output \mathcal{I} .

If there is no such x, y then let $\mathcal{J}[t+1]$ be the ordering obtained by running Procedure $P_1(\mathcal{I}[t], M)$. Continue to stage $t + 2$.

End of Procedure P_3 .

The construction (\emptyset' version).

Stage $s = 0$: Define $\mathcal{L}[0]$ to be the trivial order on $L[0] = \emptyset$ and define $M[0] = \emptyset$.

Stage $s + 1$: We are given $\mathcal{L}[s]$ and $M[s]$. Find the least $e \leq s$, if it exists, such that Q_e is *off* at the end of stage s and

$$\exists t \in \mathbb{N} \text{ such that Procedure } P_3(\mathcal{L}[s], M[s], e) \text{ terminates after } t \text{ steps.} \quad (4.4)$$

Note that whether (4.4) holds can be determined using the oracle \emptyset' .

Case 1. If there is such an e , then let $\mathcal{L}[s + 1]$ be the ordering obtained by running $P_3(\mathcal{L}[s], M[s], e)$. Let $M[s + 1] = M[s] \cup \{ \langle e, x, y \rangle \}$ where x, y is the pair plummeted by $P_3(\mathcal{L}[s], M[s], e)$ and x is e -older than y . Say that *action is taken for Q_e* at stage $s + 1$.

Case 2. If there is no such e , then let $\mathcal{L}[s + 1]$ be the ordering obtained by running Procedure $P_1(\mathcal{L}[s], M[s])$, and let $M[s + 1] = M[s]$.

End of construction.

The construction described above satisfies the *Working Hypothesis A* WHA $[s]$ from Theorem 4.1 at every stage s . The required linear extension of \mathcal{P} is $\mathcal{L} = \lim_s \mathcal{L}[s]$. It is computable in \emptyset' since the \mathcal{L} -ordering of x and y is determined at stage $s = \max\{x, y\} + 1$ at the latest. We now verify that we can either take action to satisfy each Q_e , or else Q_e is automatically satisfied.

Lemma 4.10. *Suppose that W_e is infinite and s_0 is a stage such that Q_e is off at the end of stage s_0 , and (4.4) does not hold for e at stage $s_0 + 1$. Then there are distinct $x, y \in W_e$ such that $W_e \cap (x, y)_{L^*} = \emptyset$.*

Proof. Suppose that W_e is infinite and s_0 is as stated in the lemma. Let

$$b_0 <_{L[s_0]} b_1 <_{L[s_0]} \dots <_{L[s_0]} b_{k-1}$$

be a listing of $L[s_0]$ in ascending $<_{L[s_0]}$ -order, and let b_{-1} and b_k denote $-\infty$ and $+\infty$ respectively. First, observe that for any $x \in \mathbb{N} \setminus L[s_0]$ and $0 \leq i \leq k$,

$$b_{i-1} <_L x <_L b_i \iff \left(\begin{array}{l} x \not<_P b_{i-1}, \text{ and either } x <_P b_i \text{ or} \\ b_i = x_j \text{ for some } \langle j, x_j, y_j \rangle \in M[s_0] \text{ and } x <_P y_j. \end{array} \right) \quad (4.5)$$

That is, the ultimate position of x in \mathcal{L} relative to $L[s_0]$ is computable. This can be seen by noting that Procedure P_1 preserves property (4.5), as does the application of

Procedure P_2 in conjunction with condition (iii) in Procedure P_3 . For $0 \leq i \leq k$, let

$$U_i = W_e \cap \{z \in \mathbb{N} \mid b_{i-1} <_L z <_L b_i\}.$$

The sets U_i are computably enumerable due to (4.5). Moreover, each U_i is linearly ordered by $<_P$ —as otherwise the procedure $P_3(\mathcal{L}[s_0], M[s_0], e)$ would terminate—and its $<_P$ -ordering coincides with $<_L$. Let i be such that U_i is infinite. Then computable scatteredness of \mathcal{P} implies that there is some pair $x, y \in U_i$ such that $U_i \cap (x, y)_{L^*}$ is empty. It now suffices to note that the latter is the same as $W_e \cap (x, y)_{L^*}$. \square

To finish our verification that \mathcal{L} is computably scattered for the case of $X = \emptyset'$, let s_e be the least stage such that $s_e \geq e$ and action is not taken for any $Q_{e'}$ with $e' < e$ at any stage $s \geq s_e + 1$. At stage $s_e + 1$ we either take action for Q_e (if we haven't already done so earlier), or Lemma 4.10 guarantees that Q_e is satisfied without taking action for Q_e .¹⁰

Modifications necessary for an oracle satisfying (4.3). The strategy we will use with an oracle X satisfying the escape property (4.3) is to find a \emptyset' -computable function $f(s)$ which will bound the number of steps needed for Procedure P_3 to terminate at stage s of the construction. Instead of asking the oracle directly whether (4.4) holds, we can run P_3 and see if it terminates within $f(s)$ steps.

Although the weaker oracle cannot compute f directly, it can compute a function $g(s)$ which escapes f and this will be sufficient. Instead of checking (4.4) directly, we will run P_3 for $g(s)$ steps. If it terminates within $g(s)$ steps then we proceed to satisfy Q_e as above. If not, we will assume for now that it does not terminate (and thus Q_e is satisfied automatically as in Lemma 4.10). Later it may turn out that our bound $g(s)$ was incorrect so we have to take action for Q_e at a later stage. Thus the requirements will not be satisfied in sequential order. However, we can argue that our approximation $g(s)$ will eventually bound the termination time for $P_3(\mathcal{L}[s], M[s], e)$.

Note that the role of the oracle X is limited to computing g , and thus determining which requirement will take action at each stage. The construction (for a fixed X) thus traces out a path in a finitely branching, \emptyset' -computable subtree T of $\{-1, 0, 1, 2, \dots\}^{<\omega}$ defined as follows. With each node σ of T is associated a *construction state* $(\mathcal{L}[\sigma], M[\sigma])$. The root of T is the empty string; the root node is associated with construction state $(\mathcal{L}[0], M[0])$ as defined in stage 0 of the construction. Given a string $\sigma \in T$ of length s , the string $\sigma \hat{\langle} i$ is in T if

- $0 \leq i \leq s$, and
- $\sigma(m) \neq i$ for all $m < s$ (i.e., requirement Q_i has not yet been satisfied on this branch), and
- there exists $t \in \mathbb{N}$ such that Procedure $P_3(\mathcal{L}[\sigma], M[\sigma], i)$ terminates after t steps.

In this case the construction state associated with $\sigma \hat{\langle} i$ is the pair $(\mathcal{L}[s+1], M[s+1])$ obtained by following Case 1 of the construction with $\mathcal{L}[s]$, $M[s]$ and e replaced by $\mathcal{L}[\sigma]$, $M[\sigma]$ and i , respectively. The string $\sigma \hat{\langle} -1$ is also in T , and its construction state is the pair $(\mathcal{L}[s+1], M[s+1])$ obtained by taking Case 2 of the construction with $\mathcal{L}[s]$ and $M[s]$ replaced by $\mathcal{L}[\sigma]$ and $M[\sigma]$, respectively.

¹⁰ Moreover, it is possible to show that if (4.4) holds for e at some stage $s+1$, then it also holds for e at any $s'+1 < s+1$. Therefore in fact $s_e = e$, and we either take action for Q_e at stage $e+1$ or not at all.

Note that, given a string $\sigma \in \{-1, 0, 1, 2, \dots\}^{<\omega}$, the oracle \emptyset' can compute the construction state $(\mathcal{L}[\sigma], M[\sigma])$, if $\sigma \in T$, or determine that $\sigma \notin T$. The construction with oracle \emptyset' , defined earlier, corresponds to one infinite path α through T .¹¹

We now define the \emptyset' -computable function f which will be escaped by g . For $e \in \mathbb{N}$ and a string $\sigma \in \{-1, 0, 1, 2, \dots\}^{<\omega}$, define

$$q(e, \sigma) = \begin{cases} \text{the least } t \text{ such that Procedure } P_3(\mathcal{L}[\sigma], M[\sigma], e) \\ \text{terminates in } t \text{ steps, if } \sigma \in T \text{ and if such } t \text{ exists} \\ 0 \text{ if there is no such } t, \text{ or if } \sigma \notin T. \end{cases}$$

Then q is computable from \emptyset' and $q(e, \sigma)$ bounds the time needed to determine (4.4) at step $s + 1 = |\sigma| + 1$ of a construction extending σ . However, as the construction will be done computably in X , we don't know in advance the path which the construction will take through T ; moreover it may not be computable in \emptyset' if $X \not\leq_T \emptyset'$. So to define our \emptyset' -computable search bound f we must also search over all branches of T of length s :

$$f(s) = \max_{\sigma, e} q(e, \sigma)$$

where σ ranges over all nodes of T of length s and $0 \leq e \leq |\sigma|$.

We can now give the construction for the oracle X . By (4.3), X computes a function g such that $\exists^\infty s \ g(s) \geq f(s)$. The construction for X runs exactly as specified above for oracle \emptyset' , except we replace (4.4) with the following:

$$\exists t \leq g(s) \text{ such that Procedure } P_3(\mathcal{L}[s], M[s], e) \text{ halts after } t \text{ steps.} \quad (4.6)$$

By construction, the resulting $\mathcal{L} = \lim_s \mathcal{L}[s]$ is an X -computable linear extension of \mathcal{P} . We verify that it is computably scattered. Let $W = W_e$ be an infinite c.e. set. Let s_e be the least number $\geq e$ such that action is not taken for any $Q_{e'}$ with $e' < e$ at any stage $s' \geq s_e + 1$. If there is some stage s such that (4.4) does not hold, then by Lemma 4.10 requirement Q_e is satisfied automatically. Otherwise, (4.4) holds for e at every s , so the function

$$p(s) = \text{the least } t \text{ such that } P_3(\mathcal{L}[s], M[s], e) \text{ halts after } t \text{ steps}$$

is total. Let σ_s be the node from T corresponding to the construction after stage s ; that is, σ_s is such that $\mathcal{L}[\sigma_s] = \mathcal{L}[s]$ and $M[\sigma_s] = M[s]$. Now by definition of q and f , for $s \geq e$ we have

$$p(s) = q(e, \sigma_s) \leq f(s).$$

But the function g escapes f , so there is an $s_1 \geq s_e$ such that $g(s_1) \geq f(s_1) \geq p(s_1)$. At stage $s_1 + 1$, action will be taken for Q_e (if it has not already), ensuring that Q_e is satisfied and the suborder $\mathcal{L} \upharpoonright W_e = (W_e, <_L \upharpoonright W_e)$ does not have order type η . This establishes Theorem 4.9. \square

We note again that the same construction, if applied to a computably well founded partial order \mathcal{P} , will yield a computably well founded linear extension \mathcal{L} . Thus we may replace ‘‘computably scattered’’ in the statement of Theorem 4.9 with ‘‘computably well founded’’.

¹¹By note 10, α is defined by $\alpha(s) = s$, if Case 1 was taken at stage $s + 1$, or $\alpha(s) = -1$, if Case 2 was taken at stage $s + 1$.

The property (4.3) appears to be exactly what is required for the construction above to succeed. We conjecture that the strength of the oracle can not be reduced any further.

5. NONEXTENDIBILITY RESULTS

We now show that the results in Section 4 are the best possible in the sense that, whereas for $\gamma = \omega$, a γ -c.e. linearisation can always be found that preserves computable scatteredness or computable well foundedness, this is not the case for any $\gamma < \omega$. We note that proofs of Theorems 5.1, 5.2 and 5.4 first appeared in [Gay16]. We also draw the reader's attention to the fact that the fundamental methods used for the constructions in this section are based on those used in the proofs of Theorems 1(3), 2(2) and 3 of [DHLS03].

Remark. The reader might find it instructive, before reading the Construction in the proof of Theorem 5.1, but having understood the general context presented in the preceding material—i.e. in the *Notes and Notation* etc.—to make an excursion via Theorem 5.4 whose proof affords a simplified version of some of the underlying ideas.

Theorem 5.1. *There exist a computably well founded computable partial order \mathcal{P} which has no computably well founded $\Sigma_{<\omega}^{-1}$ linear extension.*

Proof. We construct a computable partial order $\mathcal{P} = (P, <_P)$ with $P = \mathbb{N}$ as a countable disjoint union of subpartial orders $\mathcal{P}_e = (P_e, <_P \upharpoonright P_e)$ such that each \mathcal{P}_e forms a connected component of \mathcal{P} and also such that every element of \mathcal{P}_i is incomparable with every element of \mathcal{P}_j for any $i \neq j$. We will assume a computable listing $\{R_e\}_{e \in \mathbb{N}}$ —as stipulated in Section 2—of the class $\Sigma_{<\omega}^{-1}$ with associated computable approximation $\{R_e[s]\}_{e, s \in \mathbb{N}}$ defined in such a way that, for any $n \geq 1$, $R_{\langle n-1, i \rangle}$ is an n -c.e. set and $\{R_{\langle n-1, i \rangle}[s]\}_{s \in \mathbb{N}}$ is an n -c.e. approximation to $R_{\langle n-1, i \rangle}$.

For any e we will construct \mathcal{P}_e in such a way that *any* infinite chain—so *a fortiori* any ω^* sequence—in \mathcal{P}_e computes the halting set \emptyset' and is therefore not computable. Since there are no $<_P$ -comparabilities between the different components of \mathcal{P} this will imply that \mathcal{P} is computably well founded. On the other hand, for any set $R \in \Sigma_{<\omega}^{-1}$ there is an index e such that $R = R_e$ and our construction will ensure that, if (\mathbb{N}, R_e) —with R_e seen as a set of ordered pairs—is a linear extension of \mathcal{P} , then there will be a computable ω^* sequence $\{a_n\}_{n \in \mathbb{N}}$ in $(P_e, R_e \upharpoonright P_e)$.

Notation. We use the shorthand R_e linearises \mathcal{P}_e to mean that $(P_e, R_e \upharpoonright P_e)$ is a linear extension of \mathcal{P}_e and use similar shorthand relative to the approximations used during the construction, as also for R_e with regard to \mathcal{P} itself.

At each stage $s + 1$ of the construction we define an approximation $\mathcal{P}[s + 1]$ with domain some finite initial segment of \mathbb{N} . To do this we firstly compute the approximations $R_e[s + 1]$ for $e < s$ and an approximation $\emptyset'[s + 1]$ to the halting set. Starting at stage 0 with $P[0] = \emptyset$, at stage $s + 1$ we add up to four new elements to each component $\mathcal{P}_e[s]$ such that $e < s$ to obtain $\mathcal{P}_e[s + 1]$ and four new elements to the (at this point) empty component $\mathcal{P}_s[s]$ to obtain $\mathcal{P}_s[s + 1]$. When a number n enters the construction at a stage s all of the $<_P$ -comparabilities of n with numbers $k < n$ are set at stage s and do not change at any later stage. So each \mathcal{P}_e (and \mathcal{P} itself) is computable because, to decide whether a number k is in P_e it suffices to run the construction until stage $k + 1$

whereas to decide $<_P$ relative to numbers m, n , it suffices to run the construction until stage $s = \max \{m, n\} + 1$.

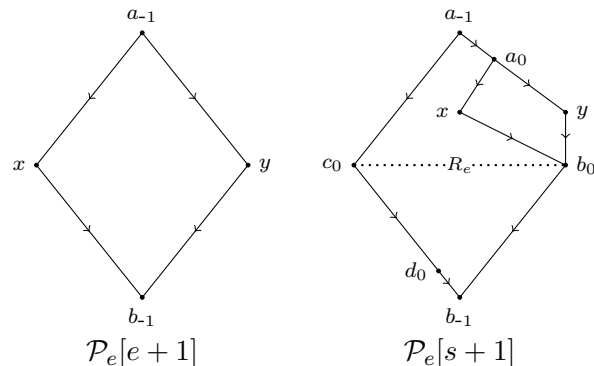


FIGURE 1. The first two stages of the construction of \mathcal{P}_e with $w <_P z$ and $w' R_e z'$ denoted by $w \leftarrow z$ and $w' \cdots R_e \cdots z'$ respectively.

Notes and Notation.

Our description will mostly relate to a single component \mathcal{P}_e of \mathcal{P} . We call a stage $s + 1 > e + 1$ *e-good* if $R_e[s + 1]$ linearises $\mathcal{P}_e[s]$. The construction of \mathcal{P}_e only proceeds at *e-good* stages. It is important to note that the set of *e-good* stages is computable.

If $s + 1$ is an *e-good* stage then, for $m, n \in P_e[s]$ we say that $m R_e n$ is *computed at stage s* if $\langle m, n \rangle \in R_e[s + 1]$. Note that, as $s + 1$ is *e-good*, $R_e[s + 1]$ linearises $\mathcal{P}_e[s]$ and so $\langle n, m \rangle \notin R_e[s]$. Notice also that for $m, n \in P_e$, $m R_e n$ is only computed at *e-good* stages by definition so that, from now on, when we say that $m R_e n$ is computed (or *r-computed* as defined below) at stage s we mean that s is an *e-good* stage. We say that $m R_e n$ is *1-computed* at stage s if it is computed at stage s and there is no earlier stage t at which $n R_e m$ is computed. For $r > 1$ we say that $m R_e n$ is *r-computed* at stage s if $m R_e n$ is computed at stage s , there is an earlier stage t such that $n R_e m$ is $r - 1$ -computed at stage t and, for any stage $t < s' < s$ and l such that $n R_e m$ is l -computed at stage s' , $l = r - 1$. Note that, if R_e is *k-c.e.* and $m R_e n$ is *r-computed* at some stage s then $r \leq k$. Notice also that it might be the case that $m R_e n$ is *1-computed* at stage s and there is an earlier stage t such that $m, n \in P_e[t - 1]$ and $\langle n, m \rangle \in R_e[t]$. In fact it might be the case that $\langle m, n \rangle \in R_e[t]$ as well. However this just means by definition that t is not an *e-good* stage so that neither $n R_e m$ nor $m R_e n$ was computed at stage t . A similar observation applies for *r-computations* with $r > 1$.

Important Shorthand Convention. To avoid cluttering notation we use the shorthand¹² “interval” to refer to a $<_P$ -interval in \mathcal{P} and—as mentioned in Section 3—drop subscripts in the notation for the latter so that, for example, (b_i, a_i) denotes the open interval with $<_P$ -lower bound b_i and $<_P$ -upper bound a_i .

We use Q_i to denote the construction’s requirement—that we also think of as a strategy—that any infinite chain in \mathcal{P} computes the question $\emptyset'(i)$ of whether i belongs to the halting set. The reader should note that Q_i can intervene in the construction of any individual component \mathcal{P}_e and at any *Level* (see later) of the construction of \mathcal{P}_e . These requirements interact (at any given Level) according to the usual process of finite

¹²This shorthand is entirely unambiguous as any interval to which we refer is always a $<_P$ -interval.

injury with Q_i being able to injure Q_j if $i < j$.

The Construction

We consider fixed index e and describe the construction of the associated component \mathcal{P}_e . Assuming that $e = \langle k - 1, i \rangle$ we know that R_e is k -c.e. and that $\{R_e[s]\}_{s \in \mathbb{N}}$ is a k -c.e. approximation to R_e . For $k > 1$ the construction has k different nested Levels. For this reason we consider three different cases according to the value of k .

Case 1: $k = 1$.

Remark. $k = 1$ implies that R_e is c.e. with $\{R_e[s]\}_{s \in \mathbb{N}}$ a c.e. approximation to R_e .

At all stages $s \leq e$ by definition $\mathcal{P}_e[s] = \emptyset$. At stage $e + 1$ four new elements a_{-1} , b_{-1} , x and y are added to $\mathcal{P}_e[e]$ to obtain $\mathcal{P}_e[e + 1]$ in such a way that $b_{-1} <_P x <_P a_{-1}$ and $b_{-1} <_P y <_P a_{-1}$ with $x \mid_P y$ as shown in Figure 1. Note that in the present case, as $k = 1$, only 1-computations are involved in the construction and so by definition everything here happens at *Level 1*. For simplicity we will omit mention of the Level at present. However the reader should keep this aspect of the construction in mind as, for the case when $k > 1$ the present case describes the Level 1 part of a construction involving k Levels. We think of the interval (a_{-1}, b_{-1}) as the *base interval* of the construction. (a_{-1}, b_{-1}) is now also the *active interval* (until the first e -good stage).

We suppose for the sake of the present argument that the set of e -good stages is nonempty (and sufficiently large). At the first such stage $s + 1 > e + 1$ we have that, for some $u, v \in \{x, y\}$, $u R_e v$ is 1-computed. Accordingly the x, y labels are removed with c_0 now labelling the element u and b_0 labelling the element v . Also four new elements a_0 , d_0 , x and y are added to $\mathcal{P}_e[s]$ to obtain $\mathcal{P}_e[s + 1]$ in such a way that $b_{-1} <_P d_0 <_P c_0$, $b_0 <_P x <_P a_0 <_P a_{-1}$, $b_0 <_P y <_P a_0 <_P a_{-1}$ and $x \mid_P y$ as shown in Figure 1. The element a_0 is now 0-activated and *painted red* and the interval (b_0, a_0) becomes the active interval of the construction. Note that the 0-activation of a_0 means that the intervals (d_0, c_0) and (b_0, a_0) have been commandeered by strategy Q_0 . The fact that a_0 is also red indicates that Q_0 has not yet registered that¹³ $0 \in \emptyset'$ and that the ongoing construction of \mathcal{P}_e will proceed entirely in the interval (b_0, a_0) for the time being. If, at some e -good stage $t + 1 > s + 1$ strategy Q_0 registers $0 \in \emptyset'$ (i.e. *receives attention*) then a_0 will be repainted blue and the construction will switch to constructing \mathcal{P}_e in the interval (d_0, c_0) .

We now consider the general case of a subsequent e -good stage $t + 1 > s + 1$. We suppose that, for some $n \geq 0$ precisely the set $\{a_0, \dots, a_n\}$ has been defined (with the associated elements b_i, c_i, d_i for each $0 \leq i \leq n$) and that there are elements $\{a_{i_0}, \dots, a_{i_h}\} \subseteq \{a_0, \dots, a_n\}$ such that a_{i_r} is r -active for each $0 \leq r \leq h$. By definition there is also an active interval which is the (smallest) interval in which the next element a_{n+1} will be defined (provided that the next e -good stage is not a \emptyset' -coding stage as defined below). Note that the active interval contains two elements labelled x, y such that $x \mid_P y$. There are two cases.

¹³For the sake of simplicity this can only happen at a subsequent e -good stage following the activation of a_0 even though it might already be the case that $0 \in \emptyset'[s + 1]$.

Notation. Strategy Q_i requires attention at e -good stage s if there is a red i -active element a_m (at the beginning of stage s) and $i \in \emptyset'[s]$. We say that s is a \emptyset' -coding stage if there is a strategy Q_i that requires attention at stage s .

Remark. To facilitate understanding of the construction we recommend a brief first reading of the Cases below combined with inspection of the ensuing example.

Case A. $t + 1$ is not a \emptyset' -coding stage. There are two subcases.

Case A(i). The active interval is (b_n, a_n) . This means that $a_{i_h} = a_n$ and that a_n is red. Also—as $t + 1$ is an e -good stage—for some $u, v \in \{x, y\}$, $u R_e v$ is 1-computed at stage $t + 1$. Accordingly the x, y labels are removed with u being relabelled c_{n+1} and v being relabelled b_{n+1} . Moreover four new elements a_{n+1}, d_{n+1}, x, y are added to $\mathcal{P}_e[t]$ to obtain $\mathcal{P}_e[t + 1]$ in such a way that $b_n <_P d_{n+1} <_P c_{n+1}$, $b_{n+1} <_P x <_P a_{n+1} <_P a_n$, $b_{n+1} <_P y <_P a_{n+1} <_P a_n$ and $x \mid_P y$, similarly to the case $n = -1$ described above. The element a_{n+1} is $h + 1$ -activated and painted red¹⁴. The active interval is now (b_{n+1}, a_{n+1}) .

Case A(ii). The active interval is (d_m, c_m) for some $0 \leq m \leq n$. Note that this means that $a_{i_h} = a_m$ and that a_m is blue; also that, for any $m \leq l \leq n$, a_l has been disactivated. Note also that each such a_l was previously j -activated for some $j > h$ but was, by definition of the construction, disactivated at the previous e -good stage $t' + 1$ (if not already disactivated) since the fact that (d_m, c_m) is the active interval implies that Q_h received attention at stage $t' + 1$. Notice also that we will be able to show that, if $n > m$, the set $\{a_{m+1}, \dots, a_n\}$ lies within the interval (b_m, a_m) and forms a descending sequence under R_e (i.e. $a_n R_e \dots R_e a_{m+1}$). Now just as in Case A(i) we have $u, v \in \{x, y\}$ such that $u R_e v$ is 1-computed at stage $t + 1$. So, as before, the x, y labels are removed with u being relabelled c_{n+1} and v being relabelled b_{n+1} . Once again four new elements a_{n+1}, d_{n+1}, x, y are added to $\mathcal{P}_e[t]$ to obtain $\mathcal{P}_e[t + 1]$ but this time in such a way that $d_m <_P d_{n+1} <_P c_{n+1}$, $b_{n+1} <_P x <_P a_{n+1} <_P c_m$, $b_{n+1} <_P y <_P a_{n+1} <_P c_m$ and $x \mid_P y$. Also, as before, the element a_{n+1} is $h + 1$ -activated and painted red. The active interval is now (b_{n+1}, a_{n+1}) .

Case B. $t + 1$ is a \emptyset' -coding stage. This means that, for some $0 \leq j \leq h$ strategy Q_j requires attention—i.e. a_{i_j} is red and $j \in \emptyset'[s + 1]$. Picking the least such j and letting $m = i_j$, Q_j now repaints a_m blue and disactivates all a_l such that $m < l \leq n$. Note that this disactivation only actually involves elements a_{i_r} such that $j < r \leq h$ since any other such a_l are already disactivated by definition of the construction. The x, y labels are removed from the two elements contained in the present active interval, and two new elements x, y are added to $\mathcal{P}_e[t]$ to obtain $\mathcal{P}_e[t + 1]$ in such a way that $d_m <_P x <_P c_m$, $d_m <_P y <_P c_m$ and $x \mid_P y$. The active interval is now (d_m, c_m) .

This ends the description of the construction. However in order to illustrate the underlying mechanics we consider a particular hypothetical example.

Example. We suppose that there are at least four e -good stages $s + 1, s_1 + 1, s_2 + 1, s_3 + 1$ and that $0 \in \emptyset'[s_2 + 1]$ whereas $0 \notin \emptyset'[s_1 + 1]$. We have that $\mathcal{P}_e[s + 1]$ is represented in Figure 1 with a_0 0-activated and red and with (b_0, a_0) now the active interval. At stage

¹⁴I.e. the strategy Q_{h+1} has commandeered the intervals (d_{n+1}, c_{n+1}) and (b_{n+1}, a_{n+1}) and now waits for further e -good stages to register (or not) that $h + 1 \in \emptyset'$.

$s_1 + 1$ Case A(i) applies resulting in $\mathcal{P}_e[s_1 + 1]$ as shown in Figure 2. a_1 is 1-activated and red and (b_1, a_1) is now the active interval. At stage $s_2 + 1$ Case B applies with Q_0 receiving attention resulting in $\mathcal{P}_e[s_2 + 1]$. Accordingly a_0 is repainted blue, a_1 is disactivated and the active interval now becomes (d_0, c_0) . At stage $s_3 + 1$ case A(ii) applies resulting in $\mathcal{P}_e[s_3 + 1]$. a_2 is now 1-activated and red and (b_2, a_2) has become the active interval.

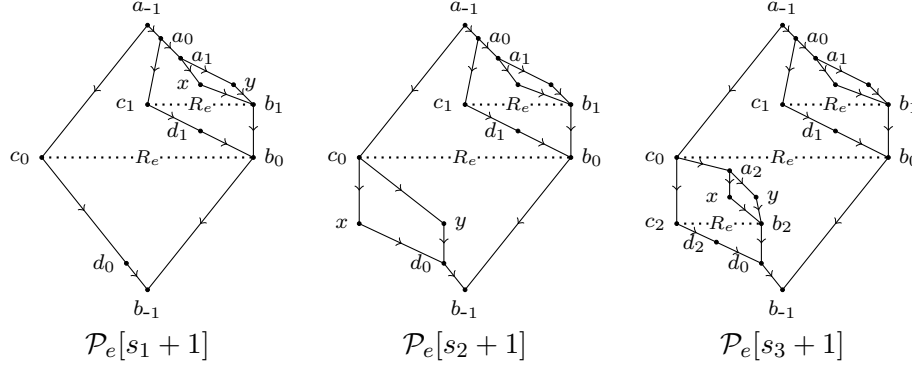


FIGURE 2. Example stages of the construction of \mathcal{P}_e with $w <_P z$ and $w' R_e z'$ denoted by $w \leftarrow z$ and $w' \cdots R_e \cdots z'$ respectively.

Note in the above example that—under the assumption that there are infinitely many e -good stages—we now have $a_2 R_e a_1 R_e a_0$ as R_e is computably enumerable. Also notice that a_0 remains permanently 0-active whereas the 1-active element was redefined once. In fact more generally it is clear by construction that, for any $i \geq 0$, after an i -active element is defined, it can be redefined up to $2^i - 1$ times. The reader might also observe that if 0 had entered \emptyset' only after several e -good stages the 1-active element would have been redefined as some a_n with $n > 2$.

Verification of Case 1.

Note firstly that if there are only finitely many e -good stages then \mathcal{P}_e will be finite and so will not embed an infinite chain. Also, an easy contrapositive argument shows us that, if R_e linearises \mathcal{P}_e , then there will be infinitely many e -good stages. We thus assume without loss of generality that there are infinitely many e -good stages.

Now consider any e -good stage s and let n_s be the greatest n such that a_{n_s} is defined at stage s . Then Case A must apply, and so a_{n_s+1} is defined, at one of the subsequent $n_s + 2$ e -good stages (as the construction “runs out” of elements to which to apply Case B). It follows by induction on n that a_n is defined for all $n \geq 0$.

Also the existence of infinitely many e -good stages and the fact that R_e is c.e. means that, for any $n \geq 0$, $c_n R_e b_n$ and moreover that this fact can be decided at the stage at which c_n and b_n are defined.

Arguing by induction over indices we see that, for each $i \geq 0$, there will be a stage t_i at which some element a_n is permanently i -activated (i.e. not disactivated at any subsequent stage). We call such an element a_n the Q_i witness.

Let $T(s + 1)$ denote the statement that, for all indices $0 \leq m < n$ and j satisfying the conjunction of conditions (i), (ii) and (iii) below, it is the case that $b_m <_P u_n$, for each $u \in \{a, b, c, d\}$.

- (i) a_m is j -active and red at the beginning of stage $s + 1$.
- (ii) a_n is defined at or before stage $s + 1$.
- (iii) a_m is still j -active at the end of stage $s + 1$ (although not necessarily red).

Then using induction over the e -good stages of the construction we deduce that, for any such stage $s + 1$, $T(s + 1)$ is true.

Now consider $n \geq 0$ and let $t + 1$ be the stage at which a_{n+1} is defined. Thus $t + 1$ is an e -good stage and a_n was already defined before the end of the previous e -good stage $s + 1$. Note that necessarily Case A of the construction applies at stage $t + 1$. If Case A(i) applies then $a_{n+1} <_P a_n$ so that *a fortiori* $a_{n+1} R_e a_n$. Also Case A(ii) only applies if Case B applied at stage $s + 1$ so that—using the notation from the construction above—either $m = n$, so that $b_m <_P a_n$ by construction, or otherwise $m < n$ in which case $b_m <_P a_n$ due to the truth of $T(s + 1)$. But then $a_{n+1} <_P c_m R_e a_n$ (as $c_m R_e b_m$) so that, once again, $a_{n+1} R_e a_n$. It follows that, if R_e linearises \mathcal{P} (and thus also its subcomponent \mathcal{P}_e), then $\{a_n\}_{n \in \mathbb{N}}$ is a ω^* sequence in $(P_e, R_e \upharpoonright P_e)$. Clearly also $\{a_n\}_{n \in \mathbb{N}}$ is computable.

Now suppose that $\mathcal{S} = (S, <_P \upharpoonright S)$ is an infinite chain in \mathcal{P}_e . Enumerate S until an element is found in either the interval $[b_0, a_0]$ or the interval $[d_0, c_0]$. (In fact in this first case we only need to enumerate three elements of S .) Note that only one of these intervals I is infinite and thus I must contain elements from S whereas, as \mathcal{S} is a chain in \mathcal{P}_e the other interval cannot contain any element of S . Note also that establishing which interval contains a member of S immediately decides $\emptyset'(0)$. Moreover this information combined with simulation of the construction allows the Q_1 witness a_n , to be computed. Then the same query made relative to the intervals $[b_n, a_n]$ and $[d_n, c_n]$ decides $\emptyset'(1)$ and allows the Q_2 witness to be computed. Continuing in this way we see that \emptyset' is computable in \mathcal{S} (and in fact in the set S itself).

Case 2: $k = 2$.

Remark. $k = 2$ implies that R_e is 2-c.e. and that $\{R_e[s]\}_{s \in \mathbb{N}}$ is a 2-c.e. approximation to R_e .

The construction proceeds as in the case $k = 1$ except that there are now 2 Levels. At stage $e + 1$, $\mathcal{P}_e[e + 1]$ is defined as in the case $k = 1$ with a_{-1}, b_{-1}, x, y being added (as in Figure 1) with the difference here that we consider these labels to be *Level 1* as also (b_{-1}, a_{-1}) to be the *Level 1 base interval* and the present *Level 1 active interval*. We also apply *Level 2* labels a'_{-1}, b'_{-1} to the elements already labelled a_{-1}, b_{-1} and consider (b'_{-1}, a'_{-1}) to be the *Level 2 base interval* and the present *Level 2 active interval*.

Note here that Level 2 is the controlling level of the construction. The Level 2 construction begins by passing control to Level 1 so that the construction now continues precisely as before for as long as there is no n such that a_n (and b_n, c_n, d_n) is defined and $b_n R_e c_n$ is 2-computed (*). (Remember that a_n, b_n, c_n, d_n are defined when $c_n R_e b_n$ is first 1-computed.) Note that it may be the case that the whole of the construction remains Level 1—i.e. if (*) occurs for no n . However, for the general case we suppose that there is a (least) e -good stage $s + 1$ such that $b_n R_e c_n$ is 2-computed for some n . Then the Level 2 construction intervenes and, choosing the pair b_n, c_n such that n is least (if there are several such pairs) relabels b_n as c'_0 and c_n as b'_0 and adds four new elements a'_0, d'_0, x, y to $\mathcal{P}_e[s]$ to obtain $\mathcal{P}_e[s + 1]$ in such a way that $b'_{-1} <_P d'_0 <_P c'_0$, $b'_0 <_P x <_P a'_0 <_P a'_{-1}$, $b'_0 <_P y <_P a'_0 <_P a'_{-1}$ and $x \upharpoonright_P y$ precisely as in Figure 1 but

with prime labels added. The Level 2 construction now abandons the preceding Level 1 definitions and all elements not bearing a Level 2 label are henceforth ignored.

Note that a'_0, d'_0 are defined to be $<_P$ -incomparable with all elements u bearing only a Level 1 label except if $u <_P b'_0$ or $c'_0 <_P u$ when the fact that $b'_0 <_P a'_0$ and $d'_0 <_P c'_0$ induces $<_P$ -comparability (due to the fact that we are constructing $<_P$ to be transitive). Since Level 2 builds in new elements to be as incomparable as possible with Level 1 elements in this way we see that the unused Level 1 elements merely leave a finite amount of “background noise” in the Level 2 construction. It is important to note that this is an ongoing feature of the Level 2 construction and so, as the latter proceeds, the unused Level 1 elements can be safely ignored without impairing the outcome of the construction.

The element a'_0 is now 0-activated and painted red and the interval (b'_0, a'_0) becomes the Level 2 active interval. Note that 0-activation of a'_0 means that the intervals (d'_0, c'_0) and (a'_0, b'_0) have been commandeered by the strategy Q_0 working at Level 2 in precisely the same way that this happened at Level 1. The Level 2 construction now resets $a_{-1} = a'_0$ and $b_{-1} = b'_0$ thereby stipulating that (a'_0, b'_0) is the new Level 1 base interval, and the ongoing Level 1 active interval, and that the Level 1 construction must start again from scratch in this interval. In this way the Level 2 construction effectively passes control back to the Level 1 construction which can be thought of as starting in a new incarnation (so that we can reuse the Level 1 labels without ambiguity). It will now only intervene in the ongoing Level 1 construction if there is an e -good stage $t+1$ at which Q_0 requires attention at Level 2 or if it finds some pair b_m, c_m (under the present reincarnated labelling) such that $b_m R_e c_m$ is 2-computed. In the first case it introduces new elements $x |_P y$ to the interval (d'_0, c'_0) , paints a'_0 blue and resets (d'_0, c'_0) to be the Level 2 active interval and, in so doing, resets $b_{-1} = d'_0$ and $a_{-1} = c'_0$. In the second case the Level 2 construction determines the least m such that $b_m R_e c_m$ is 2-computed at stage $t+1$, relabels b_m as c'_1 and c_m as b'_1 and adds new elements a'_1, d'_1, x, y in such a way that $b'_0 <_P d'_1 <_P c'_1, b'_1 <_P x <_P a'_1 <_P a'_0, b'_1 <_P y <_P a'_1 <_P a'_0$ and $x |_P y$. It paints a'_1 red and resets the Level 2 interval to be (b'_1, a'_1) and, in so doing resets $b_{-1} = b'_1$ and $a_{-1} = a'_1$. In both cases all previous Level 1 work is abandoned and the Level 2 construction passes control back to Level 1 which again restarts from scratch in the newly reassigned interval (b_{-1}, a_{-1}) .

The Level 2 construction thus proceeds in precisely the same way as the Level 1 construction with the difference that Level 1 progress is dictated by registering 1-computations on pairs supplied directly by the construction whereas Level 2 progress is dictated by registering 2-computations on pairs supplied by Level 1. In other words Cases A(i), A(ii) and B are applied in precisely the same way at Level 2 with appropriate modifications made specifying the Level 1/Level 2 interactions involved.

To verify the construction in the present case we assume as in the case $k = 1$ that there are infinitely many e -good stages. In the case that only a finite number of elements a'_n are defined we know that the Level 1 base interval (b_{-1}, a_{-1}) is either never reassigned (i.e. if not even a'_0 is defined) or eventually stabilises during the construction (as either (b'_m, a'_m) for some $m \geq -1$ or (d'_l, c'_l) for some $l \geq 0$). Hence the verification described for the case $k = 1$ applies directly to the Level 1 construction in the (eventually permanent assignment of the) interval (b_{-1}, a_{-1}) in this case. In the case when a'_n is defined for all $n \geq 0$ we can now apply the same verification but at Level 2. (Notice in particular

that now $c'_n R_e b'_n$ for all $n \geq 0$ and that this fact can be decided at the stage at which b'_n, c'_n are defined.) Accordingly we can show, just as in the verification of the Level 1 construction, that $\{a'_n\}_{n \in \mathbb{N}}$ is a computable ω^* sequence in $(P_e, R_e \upharpoonright P_e)$ and that, if $\mathcal{S} = (S, <_P \upharpoonright S)$ is an infinite chain embedded in \mathcal{P}_e , then we can compute \mathcal{O}' by querying, for $n = 0, 1, 2, \dots$, the intervals $[c'_n, d'_n]$, $[b'_n, a'_n]$ for a member of S and simulating the Level 2 construction.

Case 3: $k > 2$.

Notation. In the following we use notation of the form $a_n^{(0)}, a_n^{(1)}$ to denote the labels that we previously (for simplicity) denoted respectively a_n, a'_n (and similarly for labels involving the letters $u \in \{b, c, d\}$). Level j labels are accordingly of the form $a_n^{(j-1)}, b_n^{(j-1)}, c_n^{(j-1)}, d_n^{(j-1)}$.

Remark. R_e is k -c.e. and $\{R_e[s]\}_{s \in \mathbb{N}}$ is a k -c.e. approximation to R_e .

This is a straightforward generalisation of the Case 2 construction but now with $k > 2$ different Levels of construction. At stage $e + 1$, $\mathcal{P}_e[e + 1]$ is defined as in the case $k = 1$ (see Figure 1) with $a_{-1}^{(0)}, b_{-1}^{(0)}, x, y$ being added but this time with $a_{-1}^{(0)}, b_{-1}^{(0)}$ also being relabelled as $a_{-1}^{(j-1)}, b_{-1}^{(j-1)}$ for each $1 < j \leq k$. Accordingly, for each $1 \leq j \leq k$, under these different labellings we consider the (same) interval $(b_{-1}^{(j-1)}, a_{-1}^{(j-1)})$ to be the *Level j base interval* and ongoing *Level j active interval*.

Note that Level k is the controlling level of the construction. The Level k construction begins by passing control to Level $k - 1$ and in this way control cascades down the levels to Level 1. Hence the construction starts at Level 1 in precisely the way described in Case 2. Now consider some $1 < j \leq k$. The Level j construction waits for a (least) e -good stage $s + 1$ and Level $j - 1$ pair $b_n^{(j-2)}, c_n^{(j-2)}$ such that $b_n^{(j-2)} R_e c_n^{(j-2)}$ is j -computed. If this happens the Level j construction intervenes choosing the pair $b_n^{(j-2)}, c_n^{(j-2)}$ such that n is least and relabelling $b_n^{(j-2)}$ as $c_0^{(j-1)}$ and $c_n^{(j-2)}$ as $b_0^{(j-1)}$. It now adds four new elements $a_0^{(j-1)}, d_0^{(j-1)}, x, y$ to $\mathcal{P}_e[s]$ to obtain $\mathcal{P}_e[s + 1]$ in such a way that $b_{-1}^{(j-1)} <_P d_0^{(j-1)} <_P c_0^{(j-1)}, b_0^{(j-1)} <_P x <_P a_0^{(j-1)} <_P a_{-1}^{(j-1)}$ and also $b_0^{(j-1)} <_P y <_P a_0^{(j-1)} <_P a_{-1}^{(j-1)}$ and $x \upharpoonright_P y$. The Level j construction now abandons the preceding Level i definitions for all $1 \leq i < j$ and all elements not bearing a Level j label are henceforth ignored. The element $a_0^{(j-1)}$ is 0-activated and painted red and the interval $(b_0^{(j-1)}, a_0^{(j-1)})$ becomes the Level j active interval¹⁵. The Level j construction now resets $a_{-1}^{(i-1)} = a_0^{(j-1)}$ and $b_{-1}^{(i-1)} = b_0^{(j-1)}$ for all $1 \leq i < j$ thereby stipulating that $(a_0^{(j-1)}, b_0^{(j-1)})$ is the newly assigned Level i base interval and ongoing Level i active interval for all such i . Control gets passed back from Level j to Level $j - 1$ and hence cascaded back down to Level 1 so that each Level starts again from scratch in the interval¹⁶ $(a_0^{(j-1)}, b_0^{(j-1)})$. Thus we see that, for each Level $1 < j \leq k$ the Level j construction proceeds relative to the Level $j - 1$ precisely as described for $j = k = 2$ relative to Level 1 in Case 2 with progress of the Level j construction dictated by registering j -computations on pairs supplied by Level $j - 1$. Note however that Level j

¹⁵Note that 0-activation of $a_0^{(j-1)}$ means that the intervals $(d_0^{(j-1)}, c_0^{(j-1)})$ and $(b_0^{(j-1)}, a_0^{(j-1)})$ have been commandeered by the strategy Q_0 working at Level j .

¹⁶This interval is “seen” by each level $1 \leq i < j$ as $(a_{-1}^{(i-1)}, b_{-1}^{(i-1)})$.

proceeds under the proviso that its work can be interrupted by any Level $j < i \leq k$ causing the Level j base interval to be reset and the Level j construction having to start from scratch once again. In other words Cases A(i), A(ii) and B are applied at each Level j precisely as described for the case $k = 1$ with appropriate modification now specifying the Level j /Level $j - 1$ interaction involved, but with the further condition that in each case, for all $1 \leq i < j$ the Level i base interval is appropriately reassigned (as described above) and each such Level starts again from scratch.

To verify the construction we assume as before that there are infinitely many e -good stages. It follows that there is some $1 \leq j \leq k$ such that the definition of the Level j base interval $(b_{-1}^{(j-1)}, a_{-1}^{(j-1)})$ either is never reassigned or eventually stabilises and that, within this interval, $a_n^{(j-1)}$ is defined for all $n \geq 0$. We can now apply the verification procedure from Case 1 to the Level j construction in the interval $(b_{-1}^{(j-1)}, a_{-1}^{(j-1)})$. (Notice once again that $c_n^{(j-1)} R_e b_n^{(j-1)}$ for all $n \geq 0$ and that this fact can be decided at the stage at which $b_n^{(j-1)}, c_n^{(j-1)}$ are defined.) Accordingly we can show that $\{a_n^{(j-1)}\}_{n \in \mathbb{N}}$ is a (computable) ω^* sequence in $(P_e, R_e \upharpoonright P_e)$ and that, if $\mathcal{S} = (S, <_{\mathcal{P}} \upharpoonright S)$ is an infinite chain embedded in \mathcal{P}_e then, by querying the intervals $[c_n^{(j-1)}, d_n^{(j-1)}]$ and $[b_n^{(j-1)}, a_n^{(j-1)}]$ for a member of S for $n = 0, 1, 2, \dots$ and simulating the Level j construction, we can compute \emptyset' .

Remark. We in fact see that this verification procedure can be bundled into two algorithms, the first of which performs a search on the k -levels of the construction and, if \mathcal{P}_e is infinite, extracts the ω^* sequence in $(P_e, R_e \upharpoonright P_e)$ from the Level at which it is defined. The second algorithm can then be used to compute \emptyset' (in parallel) relative to the ω^* sequence extracted using as oracle any infinite chain \mathcal{S} embedded in \mathcal{P}_e . Note that these algorithms can be defined uniformly in $e = \langle k - 1, i \rangle$ so that for the overall partial order \mathcal{P} we have two universal algorithms performing respectively the necessary k Level search and oracle computations relative to \mathcal{P}_e for any index $e \geq 0$.

We now conclude from the work above that \mathcal{P} satisfies the statement of Theorem 5.1. Indeed, suppose that $R \in \Sigma_{<\omega}^{-1}$ linearises \mathcal{P} . Then, for some index e , $R = R_e$ and it follows that there are infinitely many e -good stages during the construction so that a computable ω^* sequence is constructed in $(P_e, R_e \upharpoonright P_e)$ as described above. Now suppose that \mathcal{S} is an infinite chain embedded in \mathcal{P} then, as the elements in different components are mutually incomparable, there is an index e such that \mathcal{S} is entirely contained in the component \mathcal{P}_e and so, as we have seen, computes \emptyset' . It follows in particular that \mathcal{P} is computably well founded. \square

Note that our construction will work for any choice of listing $\{R_e\}_{e \in \mathbb{N}}$ of the class $\Sigma_{<\omega}^{-1}$ with associated effective approximation $\{R_e[s]\}_{e, s \in \mathbb{N}}$ provided that, for each e , $\{R_e[s]\}_{s \in \mathbb{N}}$ is a k -c.e. approximation for some k . (Moreover we will still obtain two search algorithms as described in the above Remark¹⁷.)

Now suppose that our listing includes a set R_e which linearises \mathcal{P} and is ω -c.e. but is not in $\Sigma_{<\omega}^{-1}$. Suppose for the sake of argument that it is also the case that, for any $\langle u, v \rangle$, $\{R_e[s]\}_{s \in \mathbb{N}}$ “changes its mind” on $\langle u, v \rangle$ at least $\langle u, v \rangle$ times. (However there is of course by definition a computable function f such that this number of “changes of

¹⁷Our actual choice of listing $\{R_e\}_{e \in \mathbb{N}}$ simply makes the first algorithm “neater” in the sense that, given $e = \langle k - 1, i \rangle$ it knows at the start that there are k Levels involved in searching \mathcal{P}_e .

mind” is bounded by $f(\langle u, v \rangle)$.) The problem that now arises during the construction is that, for any Level j , if n is such that $a_n^{(j-1)}$ is defined and is also (large enough) such that $\langle c_n^{(j-1)}, b_n^{(j-1)} \rangle > j$, then $b_n^{(j-1)} R_e c_n^{(j-1)}$ will eventually be $j + 1$ -computed. Accordingly the construction of our putative ω^* sequence at Level j will be abandoned and the Level j construction will be restarted in a base interval newly assigned by Level $j + 1$. Since this will happen infinitely often for all j we see that the construction of an ω^* sequence as described in Theorem 5.1 breaks down. This illustrates why our construction will, as expected, not work at the ω -c.e. level.

Theorem 5.2. *There exists a (classically) scattered computable partial order \mathcal{P} which has no computably scattered $\Sigma_{<\omega}^{-1}$ linear extension.*

Proof. We construct a computable partial order \mathcal{P} as a disjoint union of subpartial orders $\mathcal{P}_e = (P_e, <_P \upharpoonright P_e)$ with no comparabilities between different components as in the proof of Theorem 5.1. We will again use the listing $\{R_e\}_{e \in \mathbb{N}}$ with its associated computable approximation $\{R_e[s]\}_{e,s \in \mathbb{N}}$. We will assume the same basic definitions and notation of the construction in the proof of Theorem 5.1. In this case however we will not be trying to code \emptyset' and so, for example, the fundamental Level 1 construction will involve only the definitions of pairs b_n, c_n (instead of the quadruples a_n, b_n, c_n, d_n defined in Theorem 5.1). Accordingly the strategies Q_i do not intervene here so that we do not need the notion of an i -active element. We will now in fact be able to talk directly about the n th active interval on any Level where for example, on Level 1, this interval is the one in which the pair b_n, c_n will be defined.

Here, given index e , we will construct \mathcal{P}_e such that, for every element $d \in P_e$, either the set $\{z \in P_e \mid z <_P d\}$ is finite or otherwise the set $\{z \in P_e \mid d <_P z\}$ is finite. Thus each individual such component \mathcal{P}_e is scattered so that, by the incomparability of elements in different components, \mathcal{P} is itself scattered. On the other hand, we will construct \mathcal{P}_e so that, if R_e linearises \mathcal{P} , then there will be a computable η sequence embedded in $(P_e, R_e \upharpoonright P_e)$.

The Construction

We consider fixed index e and describe the construction of the associated component \mathcal{P}_e . Assuming that $e = \langle k - 1, i \rangle$ we know that that $\{R_e[s]\}_{s \in \mathbb{N}}$ is a k -c.e. approximation to R_e and that, as before, if $k > 1$, then the construction has k different nested Levels. Due to the similarities with the construction of Theorem 5.1 we split our description into two cases only—the case $k = 1$ and the case $k > 1$.

Case 1: $k = 1$.

At all stages $s \leq e$ by definition $P_e[s] = \emptyset$. At stage $e + 1$ four new elements b_{-1}, c_{-1}, x and y are added to $\mathcal{P}_e[e]$ to obtain $\mathcal{P}_e[e + 1]$ in such a way that $b_{-1} <_P x <_P c_{-1}$ and $b_{-1} <_P y <_P c_{-1}$ with $x \upharpoonright_P y$ as shown in Figure 1 under the relabelling of a_{-1} and b_{-1} by c_{-1} and b_{-1} respectively. Note once again here that everything in the present ($k = 1$) case happens at Level 1 but that, for simplicity we will omit mention of the Level for the time being. The interval (b_{-1}, c_{-1}) is defined to be the 0-active interval.

We suppose for the sake of the present argument that the set of e -good stages is nonempty (and sufficiently large). At the first such stage $s + 1 > e + 1$ we have that, for some $u, v \in \{x, y\}$, $u R_e v$ is 1-computed. Accordingly the x, y labels are removed with

c_0 now labelling the element u and b_0 labelling the element v . Also two new elements x and y are added to $\mathcal{P}_e[s]$ to obtain $\mathcal{P}_e[s+1]$ in such a way that $b_{-1} <_P x <_P c_0$, $b_{-1} <_P y <_P c_0$ and $x \mid_P y$ and (b_{-1}, c_0) is now defined to be the 1-active interval.

Now consider the general case of the $n+1$ st e -good stage with $n \geq 1$. By definition the n -active interval is defined as (b_i, c_j) for some distinct $-1 \leq i, j < n$ and contains precisely the two $<_P$ -incomparable elements presently labelled x, y . We have that, for some $u, v \in \{x, y\}$, $u R_e v$ is 1-computed. Accordingly the x, y labels are removed with c_n now labelling the element u and b_n labelling the element v .

Remark. The construction is working under the assumption that b_0, \dots, b_{n-1} forms the initial segment of an η sequence under R_e and that the active n -interval was chosen so that now b_0, \dots, b_{n-1}, b_n will also form an initial segment of the sequence. Accordingly the choice of the $n+1$ -active interval is now made in such a way that when (or if) b_{n+1} is defined in this interval it also will continue the η sequence. Note that, as mentioned in Section 3, we can think of the b_i 's as labelling nodes on a binary tree with b_0 labelling the root. Under this representation the choice of the active interval at this stage is made in terms of a process of labelling the nodes of the tree of a certain length $w+1$ from left to right so as to "densify" the b_i 's which label nodes of length $\leq w$. Thus when this process has labelled the rightmost node of length $w+1$ (corresponding to action taken in an active interval of the form (b_j, c_{-1})) the process restarts at the leftmost node of length $w+2$ (corresponding to a new active interval of the form (b_{-1}, c_h)).

Now, with the above remark and the fact that the n -active interval was defined as (b_i, c_j) in mind, if $j \neq -1$ then the $n+1$ -active interval is defined to be (b_j, c_l) where c_l is the immediate R_e -successor of b_j . If $j = -1$, on the other hand, then the $n+1$ -active interval is defined to be (b_{-1}, c_h) where c_h is the immediate R_e -successor of b_{-1} . It is important to note here that in the first case c_l is the (unique) immediate $<_P$ -successor of b_j over the set $\{c_0, \dots, c_{n-1}\}$ and that likewise in the second case c_h is the immediate $<_P$ -successor of b_{-1} over this set. Thus in both cases the $n+1$ -active interval is indeed a well defined (and empty) $<_P$ -interval. Also note that, by construction, b_j is the immediate R_e -successor of c_j . (The properties assumed here can be easily checked during the induction argument over the e -good stages of the construction stipulated below.) Using (b_i, c_j) to denote the $n+1$ -active interval, two new elements x and y are added to $\mathcal{P}_e[s]$ to obtain $\mathcal{P}_e[s+1]$ in such a way that $b_i <_P x <_P c_j$, $b_i <_P y <_P c_j$ and $x \mid_P y$.

To illustrate what is happening note that, at the end of the third e -good stage $s+1$, $\mathcal{P}_e[s+1] \setminus \{x, y\}$ is as follows,

$$b_{-1} <_P c_1 R_e b_1 <_P c_0 R_e b_0 <_P c_2 R_e b_2 <_P c_{-1}$$

with the 3-active interval being (b_{-1}, c_1) and the elements x, y inserted into this interval. Note also that $b_1 R_e b_0 R_e b_2$. Continuing further we find at the end of the seventh e -good stage that

$$b_3 R_e b_1 R_e b_4 R_e b_0 R_e b_5 R_e b_2 R_e b_6$$

and we see that the construction has now enumerated the first 7 elements of an η sequence (under R_e).

Verification of Case 1.

We assume again without loss of generality—see the proof of Theorem 5.1—that there are infinitely many e -good stages.

Then, by induction over e -good stages we easily check that, for all $n \geq 0$, at the end of the $n + 1$ st e -good stage the construction has enumerated the elements b_0, \dots, b_n ordered under R_e as the initial segment of an η sequence. Therefore our assumption that there are infinitely many e -good stages implies that the construction computes $\{b_n\}_{n \in \mathbb{N}}$ as an η sequence (and so dense chain) under R_e .

For all $m \geq 0$, define C_m to be the set of elements $u \in \mathcal{P}_e[\hat{s} + 1]$ such that $c_m <_P u$ with $\hat{s} + 1$ being the $m + 1$ st e -good stage, i.e. the stage when c_m is defined. We now show by induction over e -good stages that, for any such m , and e -good stage $s + 1$ such that c_m has been defined before the end of this stage, (i) and (ii) below hold.

- (i) $C_m \subseteq \{c_{-1}, \dots, c_{m-1}\}$,
- (ii) For all $u \in \mathcal{P}_e[s + 1]$, $c_m <_P u \Leftrightarrow u \in C_m$.

Indeed at the first e -good stage we have $b_{-1} <_P c_0 <_P c_{-1}$ and $b_{-1} <_P b_0 <_P c_{-1}$ and no other $<_P$ -comparabilities. Therefore both (i) and (ii) hold with $C_0 = \{c_{-1}\}$. Now at the (beginning of the) $n + 1$ st e -good stage $s + 1$ there exist $-1 \leq i, j < n$ such that (b_i, c_j) is the n -active interval with $x, y \in (b_i, c_j)$ and such that the construction relabels these elements as b_n, c_n so that $c_n R_e b_n$. Accordingly we have under this new labelling $b_i <_P b_n <_P c_j$ and $b_i <_P c_n <_P c_j$ and no other new comparabilities introduced beyond those dictated by the transitivity of R_e . By the induction hypothesis relative to the n th e -good stage $t + 1$, for each $0 \leq h < n$ the set C_h is defined such that $C_h \subseteq \{c_{-1}, \dots, c_{h-1}\}$ and such that, for all $u \in \mathcal{P}_e[t + 1]$, $c_h <_P u \Leftrightarrow u \in C_h$. This implies that, as $b_i \notin C_h$, for all $u \in \mathcal{P}_e[s + 1]$, we also have $c_h <_P u \Leftrightarrow u \in C_h$. On the other hand clearly, for $u \in \mathcal{P}_e[s + 1]$, $c_n <_P u \Leftrightarrow u \in C_n$ where we define $C_n = \{c_j\} \cup C_j$. Note that our induction hypothesis (relative to C_h for $h < n$) and the fact that $j < n$ implies that $C_n \subseteq \{c_{-1}, \dots, c_{n-1}\}$. We thus conclude that the statement above is true. Note that this means that, for any $n \geq 0$, the set of elements $<_P$ -above c_n in \mathcal{P}_e is the finite set C_n . A similar argument shows that, for any $n \geq 0$ the set of elements $<_P$ -below b_n in \mathcal{P}_e is finite. Thus \mathcal{P}_e is scattered.

Case 2: $k > 1$.

Notation. We use notation of the form $b_n^{(0)}, c_n^{(0)}$ to denote the labels that we previously denoted respectively b_n, c_n . Level j labels are accordingly of the form $b_n^{(j-1)}, c_n^{(j-1)}$.

This is a straightforward application of Cases 2 and 3 in the proof of Theorem 5.1 to the context of the present Level 1 construction. In other words the construction now has k Levels and each Level $j > 1$ processes its m -active interval by searching the Level $j - 1$ construction in this interval for a Level $j - 1$ pair $b_n^{(j-2)}, c_n^{(j-2)}$ such that $b_n^{(j-2)} R_e c_n^{(j-2)}$ is j -computed (whereas the Level $j - 1$ labels have been in place since $c_n^{(j-2)} R_e b_n^{(j-2)}$ was first $j - 1$ computed). When (or if) such a pair is found the Level j construction chooses the pair with least index n and relabels $c_n^{(j-2)}$ as $b_m^{(j-1)}$ and $b_n^{(j-2)}$ as $c_m^{(j-1)}$ (so that we now have $c_m^{(j-1)} R_e b_m^{(j-1)}$) chooses its $m + 1$ -active interval $(b_i^{(j-1)}, c_l^{(j-1)})$ in precisely the way that this is done at Level 1, and inserts two new elements x, y such that $x |_P y$ in this interval. It then, for all $1 \leq r < j$, resets the Level r 0-active interval to its $m + 1$ -active interval by resetting $b_{-1}^{(r-1)} = b_i^{(j-1)}$ and

$c_{-1}^{(r-1)} = c_i^{(j-1)}$ thus causing Level r to abandon all previous work and signalling that it is to restart in this interval. Level j then passes control back to Level $j - 1$ so that the control cascades down to Level 1 and the construction restarts from scratch at all Levels $r < j$ in this interval.

The Level j construction thus proceeds precisely as described for Level 1 with the difference that it registers j -computations of pairs supplied by the Level $j-1$ —instead of the process of registering 1-computations of pairs supplied directly by the construction as happens at Level 1—under the proviso that its own work may be interrupted by the action of some Level t with $j < t \leq k$ in a similar way to that just described. It is important to note that, whenever Level j intervenes all elements of the construction that do not have a Level j label are abandoned. (Note here that an element has Level r label with $j < r \leq k$ only if it already has a Level j label.) Thus, for example, when Level j intervenes for the first time the only elements now relevant to the construction are $b_{-1}^{(j-1)}, c_{-1}^{(j-1)}, b_0^{(j-1)}, c_0^{(j-1)}$ and the newly chosen elements x, y (with $b_{-1}^{(j-1)}, c_{-1}^{(j-1)}$ being the elements that were also originally labelled $b_{-1}^{(0)}, c_{-1}^{(0)}$ whereas $b_0^{(j-1)}, c_0^{(j-1)}$ is the relabelling of some pair $c_n^{(j-2)}, b_n^{(j-2)}$ as described above but with $m = 0$). Note that the abandoned elements of lower Levels once again cause a certain amount of “background noise” at Level j . However, since no $<_P$ -comparabilities between new elements and abandoned elements are defined (beyond those dictated by the fact that $<_P$ must be transitive) this “background noise” causes only finite interference to the work carried out at Level j and can thus be safely ignored.

To verify the construction for $k > 1$ we assume as before that there are infinitely many e -good stages. It follows that there is some $1 \leq j \leq k$ such that the definition of the Level 1 base interval $(b_{-1}^{(j-1)}, c_{-1}^{(j-1)})$ either is never reassigned or eventually stabilises and that, within this interval $a_n^{(j-1)}$ is defined for all $n \geq 0$. We can now apply the Case 1 verification procedure to the Level j construction in the interval $(b_{-1}^{(j-1)}, c_{-1}^{(j-1)})$. (Notice that $c_n^{(j-1)} R_e b_n^{(j-1)}$ for all $n \geq 0$ and that this fact can be decided at the stage at which $b_n^{(j-1)}, c_n^{(j-1)}$ are defined.) Accordingly we can show that $\{b_n^{(j-1)}\}_{n \in \mathbb{N}}$ is a (computable) η sequence in $(P_e, R_e \upharpoonright P_e)$ and that every element in \mathcal{P}_e has either at most finitely elements $<_P$ -above it at most finitely many elements $<_P$ below it. (Here we take into account the abandoned elements of the construction as well as the elements $b_n^{(j-1)}, c_n^{(j-1)}$.)

Remark. Similarly to before we see that the construction intrinsic to the verification procedure can be bundled into a single algorithm, which performs a search on the k -levels of the construction and, if \mathcal{P}_e is infinite, extracts the η sequence in $(P_e, R_e \upharpoonright P_e)$ from the Level at which it is defined. Note that this algorithm can be defined uniformly in $e = \langle k - 1, i \rangle$ so that for the overall partial order \mathcal{P} we have a universal algorithm performing the necessary k Level search relative to \mathcal{P}_e for any index $e \geq 0$.

We now conclude from the work above that \mathcal{P} satisfies the statement of Theorem 5.2. Indeed, suppose that $R \in \Sigma_{<\omega}^{-1}$ linearises \mathcal{P} . Then for some index e , $R = R_e$ and it follows that there are infinitely many e -good stages during the construction so that a computable η sequence is constructed in $(P_e, R_e \upharpoonright P_e)$ as described above. Now as we have seen, for any index e the component \mathcal{P}_e is scattered. Thus, as elements from different components are pairwise incomparable we see that \mathcal{P} is itself scattered. \square

Note 5.3. Suppose that α is a computable order type such that $\zeta \leq \alpha$. Then the computable partial order \mathcal{P} constructed in Theorem 5.2 does not embed α whereas any $\Sigma_{<\omega}^{-1}$ linearisation \mathcal{L} of \mathcal{P} computably embeds α (as α computably embeds into the η sequence that is constructed in \mathcal{L}).

To end this section we turn our attention to the order type of the integers ζ . We firstly note that, given a computable partial order $\mathcal{P} = (\mathbb{N}, <_P)$ which does not computably embed ζ , there is a Δ_5^0 linearisation of \mathcal{P} that does not computably embed ζ . Indeed let G_0 be the set of numbers a such that, for any index e , if $W_e \subseteq \{z \in \mathbb{N} \mid z <_P a\}$, then W_e does not define an ω^* sequence in \mathcal{P} . Then G_0 is Π_3^0 . Also, letting $G_1 = \mathbb{N} \setminus G_0$ (so that G_1 is Σ_3^0) we see that, as \mathcal{P} does not computably embed ζ , for any $b \in G_1$ and all indices e , if $W_e \subseteq \{z \in \mathbb{N} \mid b <_P z\}$, then W_e does not define an ω sequence in \mathcal{P} . Moreover, for any $a \in G_0$ and $b \in G_1$ it is not the case that $b <_P a$. Hence, as both G_0 and G_1 are $\emptyset^{(3)}$ -computable, using our construction in the proof of Theorem 4.8 (or Rosenstein's construction proving Theorem 2.1), relativised to $\emptyset^{(3)}$ we obtain $\emptyset^{(4)}$ -computable linearisations \mathcal{L}_0 and \mathcal{L}_1 of $(G_0, <_P \upharpoonright G_0)$ and $(G_1, <_P \upharpoonright G_1)$ respectively such that \mathcal{L}_0 does not computably embed ω^* and \mathcal{L}_1 does not computably embed ω . Thus $\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_1$ is a Δ_5^0 linearisation of \mathcal{P} that does not computably embed ζ . So the question that now arises is whether, using more delicate arguments, we can find such a linearisation \mathcal{L} which is Δ_2^0 or even ω -c.e. We now answer this question in the negative.

Theorem 5.4. *There exists a computable partial order \mathcal{P} which does not embed ζ such that any Δ_2^0 linearisation of \mathcal{P} computably embeds ζ .*

Proof. For the present proof we assume that $\{R_e\}_{e \in \mathbb{N}}$ is a listing of the class of Σ_2^0 sets with associated effective Σ_2^0 approximation $\{R_e[s]\}_{e, s \in \mathbb{N}}$. In other words—using in addition our identification of numbers with pairs coded by $\langle \cdot, \cdot \rangle$ —we have that, for all indices e ,

$$R_e = \{ \langle n, m \rangle \mid \exists t (\forall s \geq t) [R_e(\langle n, m \rangle)[s] = 1] \}$$

and that, for every Σ_2^0 set R , there is some e such that $R = R_e$.

We construct a computable partial order \mathcal{P} as a disjoint union of subpartial orders \mathcal{P}_e with no $<_P$ -comparabilities between different components as in the proofs of Theorem 5.1 and 5.2 with the main difference that this time we use the Σ_2^0 approximation $\{R_e[s]\}_{e, s \in \mathbb{N}}$. Again we assume the basic definitions of our earlier proofs but with an ingredient of simplification which will be described below.

Given index e we will construct $\mathcal{P}_e = (P_e, <_P \upharpoonright P_e)$ such that, for every chain \mathcal{L} in \mathcal{P}_e , if \mathcal{L} is not finite, then it has order type either ω^* or ω . Our construction will also ensure that, if (\mathbb{N}, R_e) linearises \mathcal{P} then there will be a computable copy of ζ embedded in $(P_e, R_e \upharpoonright P_e)$.

The Construction

We consider fixed index e and describe the construction of the associated component \mathcal{P}_e . At all stages $s \leq e$ by definition $\mathcal{P}_e[s] = \emptyset$. At stage $e+1$ two new elements x and y are added to $\mathcal{P}_e[e]$ to obtain $\mathcal{P}_e[e+1]$ in such a way that $x \mid_P y$. We say that $s+1$ is an e -good¹⁸ stage during the present construction if $R_e[s+1]$ linearises the subcomponent

¹⁸If we choose $\{R_e\}_{e \in \mathbb{N}}$ and its Σ_2^0 approximation such that, for every Σ_2^0 (so in particular every Δ_2^0) set R there is an index e such that $R = R_e$ and the approximation $\{R_e[s]\}_{s \in \mathbb{N}}$ has infinitely many *good* stages—i.e. stages s such that $R_e[s] \subseteq R_e$ —then we can apply the definition of e -good stage from our

of \mathcal{P}_e comprising these two $<_P$ -incomparable elements (whatever the ongoing labelling of the elements). Note that, as before, the set of e -good stages is computable.

We suppose, for the sake of the present argument, that the set of e -good stages is nonempty (and sufficiently large). At the first e -good stage $s + 1 > e + 1$ we have that, for some $u, v \in \{x, y\}$, $u R_e v$ is 1-computed. We now remove the x, y labels and label the element u as c and v as b . We also add two new elements a_0 and d_0 to $\mathcal{P}_e[s]$ to obtain $\mathcal{P}_e[s + 1]$ such that $d_0 <_P c$ and $b <_P a_0$ (with no other $<_P$ -comparabilities introduced). Note that the idea is that d_0 is to be the second element of an ω^* sequence in \mathcal{P}_e and a_0 is to be the second¹⁹ element of an ω sequence in \mathcal{P}_e . Accordingly at each subsequent e -good stage for as long as $b R_e c$ is not 2-computed—i.e. $c R_e b$ is 1-computed at every such stage—the construction adds two new elements so as to continue building an ω^* sequence $<_P$ -below c and an ω sequence $<_P$ -above b in \mathcal{P}_e . Thus for $n > 0$, if $c R_e b$ is 1-computed at the $n + 1$ st e -good stage $s + 1$, then $\mathcal{P}_e[s]$ is made up of the two chains $d_{n-1} <_P \cdots <_P d_0 <_P c$ and $b <_P a_0 <_P \cdots <_P a_{n-1}$ and two new elements d_n, a_n are added so that $\mathcal{P}_e[s + 1]$ is made up of $d_n <_P d_{n-1} <_P \cdots <_P d_0 <_P c$ and $b <_P a_0 <_P \cdots <_P a_{n-1} <_P a_n$.

In the spirit of our earlier proofs we say that the construction is Level 1 for as long as $b R_e c$ is not 2-computed. Note that clearly, if there are infinitely many e -good stages and the construction remains permanently at Level 1, then \mathcal{P}_e will consist of a copy of ω^* with first element c and a copy of ω with first element b (with no comparabilities between pairs of elements in the two different chains). If—as will generally be the case—there is an e -good stage at which $b R_e c$ is 2-computed then the construction abandons the Level 1 construction and moves to Level 2 which means that it starts building—from the first such e -good stage onwards—an ω^* sequence $<_P$ -below b and an ω sequence $<_P$ -above c in \mathcal{P}_e . Level 2 now continues constructing these sequences at all e -good stages for as long $c R_e b$ is not 3-computed at any such stage. If $c R_e b$ is eventually 3-computed the construction abandons the Level 2 construction and moves to Level 3 which starts from scratch building an ω^* sequence $<_P$ -below c , whose elements are $<_P$ -incomparable with the (finite) Level 1 chain previously built $<_P$ -below c , and an ω sequence $<_P$ -above b , whose elements are likewise $<_P$ -incomparable with the Level 1 chain previously built $<_P$ -above b .

Overall the construction proceeds in Levels in a similar way to that outlined above for Levels 1-3 with the construction always working at some specific Level $k \geq 1$. Now supposing without loss of generality that k is odd—noting that the case when k is even is the same with the roles of b and c inverted—the construction remains at Level k , provided that at every intervening e -good stage $c R_e b$ is k -computed. Meanwhile the construction proceeds to build (at each e -good stage) an ω^* sequence $<_P$ -below c , whose elements are $<_P$ -incomparable with all elements belonging to chains previously built $<_P$ -below c , and an ω sequence $<_P$ -above b , whose elements are likewise $<_P$ -incomparable with all elements belonging to chains previously built $<_P$ -above b . (Notice that the chains concerned are those built at Levels i with $i < k$ and i odd.) If however $b R_e c$ is $k + 1$ computed at some subsequent e -good stage, then the construction moves

earlier proofs. However since this approach is not necessary here, for simplicity we apply the above modified version of e -goodness.

¹⁹From the point of view of Note 5.5 below d_0 and a_0 are respectively the *first* elements of the ω^* and ω sequences.

to Level $k + 1$ and starts from scratch building sequences $<_P$ -below b and $<_P$ -above c satisfying the same (incomparability) conditions as those just described with k being replaced by $k + 1$ and b and c swapped.

Verification.

We assume again without loss of generality, as in the proofs of Theorem 5.1 and 5.2—that there are infinitely many e -good stages (otherwise \mathcal{P}_e is finite and it cannot be the case that R_e linearises \mathcal{P}). Notice that elements b and c are permanently defined from the first e -good stage onwards. There are two cases. The first is the case in which the construction processes Level k for all $k \geq 1$. Note that this means that there will be infinitely many pairwise $<_P$ -incomparable finite chains built $<_P$ -below b and infinitely many pairwise $<_P$ -incomparable finite chains built $<_P$ -above b and likewise for c in place of b . Moreover any maximal chain either includes b and is made up of b added to one of the finite chains built $<_P$ -below b and to one of the finite chains built $<_P$ -above b , or otherwise includes c and has the same description with c swapped for b . I.e. any such chain is finite. The second case is when there is a Level k such that, from some stage onwards the construction remains at Level k . Then, supposing without loss of generality that k is odd, \mathcal{P}_e is made up of an ω^* sequence with c as first element and an ω sequence with b as first element—with no $<_P$ -incomparabilities between members of the different sequences—and a finite part corresponding to action taken at levels $i < k$ (i.e. the “background noise” of the construction of \mathcal{P}_e). It follows therefore that in both cases \mathcal{P}_e does not embed ζ . Moreover since the present description applies to all indices e and the subcomponents \mathcal{P}_e of \mathcal{P} are pairwise $<_P$ -incomparable we see that \mathcal{P} itself does not embed ζ .

Now suppose that R_e linearises \mathcal{P} . Suppose also, without loss of generality, that $b R_e c$. Then there is a stage $t_e > e + 1$ such that, for all $s \geq t_e$, $R_e(\langle b, c \rangle)[s] = 1$. Moreover, as R_e linearises \mathcal{P} and $\{R_e[s]\}_{s \in \mathbb{N}}$ is a Σ_2^0 approximation to R_e we know that for infinitely many stages $s \geq t_e$, $R_e(\langle c, b \rangle)[s] = 0$. However clearly each one of these stages is an e -good stage of the construction. Moreover the construction is at some fixed level k at all such stages (where $k - 1$ is in effect the total number of “changes of mind” of R_e relative to the pair b, c at previous e -good stages of the construction.) Thus, as already mentioned above, c is the first element of an ω^* sequence and b is the first element of an ω sequence in \mathcal{P}_e . Also, since $c R_e b$ and R_e linearises \mathcal{P}_e we have a copy of ζ in $(P_e, R_e \upharpoonright P_e)$. Clearly this copy of ζ is computable by construction. This concludes the proof of Theorem 5.4 given that, for any Δ_2^0 set R , there is some e such that²⁰ $R = R_e$. □

Note 5.5. Let α and β be infinite computable order types for which there exist, respectively, computable copies \mathcal{L}_α^+ and \mathcal{L}_β^- of $\alpha + 1$ and $1 + \beta$ such that neither \mathcal{L}_α^+ nor

²⁰Note that any R_e that linearises $\mathcal{P} = (\mathbb{N}, <_P)$ is Δ_2^0 . This is a special case of the fact (already mentioned in the unrelativised case in the Introduction) that, for any set A , any linear order which is either A -c.e. or A -co-c.e. but which has A -computable domain is in fact A -computable. (In the present case R_e is A -c.e. for $A = \emptyset'$.) Note also that we can also easily prove that R_e is Δ_2^0 directly in this case by including an analysis of the *ages* of both the element $\langle n, m \rangle$ and $\langle m, n \rangle$ in the approximation $\{R_e[s]\}_{s \in \mathbb{N}}$. Notice moreover that we could assume our (choice of) listing $\{R_e\}_{s \in \mathbb{N}}$ to be such that, if R is Δ_2^0 then there is an index such that $R = R_e$ with $\{R_e[s]\}_{s \in \mathbb{N}}$ being a Δ_2^0 approximation. We do not make this assumption as it does not simplify our argument.

\mathcal{L}_β^- computably embeds the order type $\alpha + \beta$. (In particular, this is obviously the case when both $\alpha + \beta \not\leq \alpha + 1$ and $\alpha + \beta \not\leq 1 + \beta$.) Then by modifying the construction in the proof of Theorem 5.4 to construct \mathcal{L}_α^+ instead of a copy of ω^* ($= \omega^* + 1$) and \mathcal{L}_β^- instead of a copy of ω ($= 1 + \omega$) we see that, for any such α and β , there exists a computable partial order \mathcal{P} which does not computably embed $\alpha + \beta$ such that any Δ_2^0 linearisation of \mathcal{P} computably embeds $\alpha + \beta$. Moreover, if in fact $\alpha + \beta \not\leq \alpha + 1$ and $\alpha + \beta \not\leq 1 + \beta$, then the first “computably” in this statement can be dropped. Thus, for example, we could replace ζ by the order type $\omega + \omega^*$ in the statement of Theorem 5.4.

Note 5.6. Suppose that in the construction of \mathcal{P} in the proof of Theorem 5.4, for all $e \geq 0$, we use the component \mathcal{P}_{2e} (instead of \mathcal{P}_e) to “diagonalise” against R_e —i.e. to ensure that, if R_e linearises \mathcal{P} , then $(P_{2e}, R_e \upharpoonright P_{2e})$ computably embeds ζ —and that we use component \mathcal{P}_{2e+1} to code $\emptyset'(e)$ as in the proof of Claim 1 of Theorem 3 in [DHLS03] (choosing f such that $\text{Ran } f = \emptyset'$). Note that we can do this as follows. We name the first two elements to be put into \mathcal{P}_{2e+1} as x_e, y_e (instead of using the temporary labels x, y as in the above proof). Then at every subsequent stage, for as long as we see that e has not entered \emptyset' , we add two new elements in such a way as to progressively construct a copy of $\omega^* <_P$ -below x_e and a copy of $\omega <_P$ -above y_e in \mathcal{P}_{2e+1} . However if we see that e enters \emptyset' then we switch to building a copy of $\omega^* <_P$ -below y_e and a copy of $\omega <_P$ -above x_e in \mathcal{P}_{2e+1} . Then we find that any linearisation $\mathcal{L} = (\mathbb{N}, <_L)$ of \mathcal{P} which does not computably embed ζ is not only *not* Δ_2^0 but also computes the halting set \emptyset' since $\emptyset' = \{e \mid x_e <_L y_e\}$. In other words this variant of the construction forces the Turing degree of such \mathcal{L} to be strictly above $\mathbf{0}'$, the Turing degree of \emptyset' . Observe also that Note 5.5 can again be applied to this result.

REFERENCES

- [Ars11] M.M. Arslanov. The Ershov Hierarchy. In S.B. Cooper and A. Sorbi, editors, *Computability in Context: Computation and Logic in the Real World*, pages 49–100. World Scientific, 2011.
- [Bon69] Robert Bonnet. Stratifications et extension des genres de chaines dénombrables. *CR Acad. Sci. Paris Sér. AB*, 269:A880–A882, 1969.
- [BP82] Robert Bonnet and M Pouzet. Linear extensions of ordered sets. In *Ordered sets*, pages 125–170. Springer, 1982.
- [Coo04] S.B. Cooper. *Computability Theory*. CRC Mathematics Series. Chapman and Hall, 2004.
- [DHLS03] R.G. Downey, D.R. Hirschfeldt, S. Lempp, and R Solomon. Computability-theoretic and proof-theoretic aspects of partial and linear orderings. *Israel Journal of Mathematics*, 138:271–289, 2003.
- [Dow98] R. Downey. Computability theory and linear orderings. In Y.L. Ershov, S.S. Goncharov, A. Nerode, and J.B. Remmel, editors, *Handbook of Recursive Mathematics Volume 2: Recursive Algebra, Analysis and Combinatorics*, Studies in Logic and the Foundations of Mathematics, pages 823–976. North Holland, 1998.
- [Ers68a] Y.L. Ershov. A certain hierarchy of sets I. *Algebra and Logic*, 7(1):47–74, 1968.
- [Ers68b] Y.L. Ershov. A certain hierarchy of sets II. *Algebra and Logic*, 7(4):15–47, 1968.
- [Ers70] Y.L. Ershov. A hierarchy of sets III. *Algebra and Logic*, 9:34–51, 1970.
- [Gay16] J. Gay. *Computably Extendible Order Types*. PhD thesis, The University of Leeds, UK, September 2016. (Available online at <http://etheses.whiterose.ac.uk/13976>).
- [Lee11] K.I. Lee. *Automorphisms and Linearisations of Computable Orderings*. PhD thesis, The University of Leeds, UK, September 2011. (Available online at <http://etheses.whiterose.ac.uk/2166>).
- [Nie09] André Nies. *Computability and randomness*. Oxford University Press, Oxford, New York, 2009.

- [Odi89] P. G. Odifreddi. *Classical Recursion Theory*. Elsevier Science B.V., Amsterdam, 1989.
- [Ros84] Joseph G. Rosenstein. Recursive linear orderings. *North-Holland Mathematics Studies*, 99:465–475, 1984.
- [Soa87] R.I. Soare. *Recursively Enumerable Sets and Degrees*. Springer Verlag, Berlin, 1987.
- [Web12] R. Weber. *Computability Theory*. American Mathematical Society, 2012.