ON THE COMPLEXITY OF THE SUCCESSIVITY RELATION IN COMPUTABLE LINEAR ORDERINGS

ROD DOWNEY, STEFFEN LEMPP, AND GUOHUA WU

Abstract. In this paper, we solve a long-standing open question (see, e.g., Downey [6, §7] and Downey and Moses [11]), about the spectrum of the successivity relation on a computable linear ordering. We show that if a computable linear ordering $\mathcal{L}$ has infinitely many successivities, then the spectrum of the successivity relation is closed upwards in the computably enumerable Turing degrees. To do this, we use a new method of constructing Δ^0_3-isomorphisms, which has already found other applications such as Downey, Kastermans and Lempp [9] and is of independent interest. It would seem to promise many further applications.

1. Introduction and Main Theorem

This paper falls into part of a long-term program in computable model theory, where we study the *spectrum* problem for relations on various classes of computable models. Suppose R is a relation on a computable model $\mathcal{A}$. If $\widehat{\mathcal{A}}$ is a computable model isomorphic to $\mathcal{A}$, then we will let $\widehat{R}$ denote the image of R in $\widehat{\mathcal{A}}$. Then the *spectrum of* R is defined as the collection of Turing (or perhaps other) degrees of $\widehat{R}$ as we run over all computable $\widehat{\mathcal{A}} \cong \mathcal{A}$.

Our understanding of possible degree spectra for distinguished relations on various structures has seen significant advances in recent years. We know that for many relations, the typical spectra we would expect to see would consist either of a single element or of infinitely

2000 *Mathematics Subject Classification.* 03C57.

Key words and phrases. computable linear order, successivity.

Downey's research was supported by a Marsden grant. Lempp's research was supported by NSF grants DMS-0140120 and DMS-0555381 as well as Grant # 13407 by the John Templeton Foundation entitled "Exploring the Infinite by Finitary Means". Wu's research was supported by a research grant RG58/06 from Nanyang Technological University. Lempp and Wu would like to thank Victoria University of Wellington for its hospitality during the time this work was carried out. The authors would also like to thank Frolov for pointing out an error in an earlier version of this paper.

many elements. (See, e.g., Harizanov [16, 17], Hirschfeldt [19], Goncharov [15], or Moses [24].) We have also made great strides in the construction of models such as graphs, partial orderings, etc., with many pathological spectra such as 2-element spectra. (See, e.g., Harizanov [18], Hirschfeldt [19], Hirschfeldt, Khoussainov, Shore and Slinko [20].) These results work by using families of computable functions and coding or "special component" arguments (which can be viewed as the being the same), such as in Hirschfeldt's Ph.D. thesis. This method usually gives graphs and then, if there is enough structure, it is possible to code these graphs into computable structures such as groups, rings, partial orderings and the like. We remark that the concern of the present paper is computable linear orderings, where it seems much more difficult to achieve such rigid degree control.

However, for *particular* relations and *particular* structures, our understanding is much more limited. In the present paper, we will be concerned with a basic relation on a linear ordering, namely the *successivity* (or *adjacency*) relation $\mathrm{Succ}(\mathcal{L})$. Here $\mathrm{Succ}(\mathcal{L})$ holds for a pair of elements (x, y) in a linear ordering $\mathcal{L} = (L, <_{\mathcal{L}})$ iff $x <_{\mathcal{L}} y$ and there is no $z \in L$ with $x <_{\mathcal{L}} z <_{\mathcal{L}} y$. We already know that the successivity relation on computable linear orderings is important in terms of its algorithmic properties. For example, we have the following

Theorem 1.1 (Dzgoev and Goncharov [12], Remmel [26])**.** *A computable linear ordering is computably categorical*[1] *if and only if it has only finitely many successivities.*

What can we say about the spectrum of the successivity relation of a particular computable linear ordering?

If the linear ordering has only finitely many successivities then this question is plainly trivial. Thus, henceforth, we will only consider linear orderings with *infinitely many* successivities.

Clearly, if $\mathcal{L}$ is a computable linear ordering, then $\mathrm{Succ}(\mathcal{L})$ is a co-c.e. set (i.e., the complement of a computably enumerable set) of pairs, and hence the possible degree spectra will be subsets of the computably enumerable (c.e.) degrees. What subsets of the c.e. degrees can be realized as possible spectra? It is known that the following spectra are possible:

(i) All c.e. degrees. (This is true by general results of Goncharov [15] for the case when $\mathcal{L}$ has a copy with $\mathrm{Succ}(\mathcal{L})$ computable and infinite, though it is easy to prove directly.)

[1]Recall that a computable structure $\mathcal{A}$ is called *computably categorical* iff for all computable structures $\mathcal{B} \cong \mathcal{A}$, $\mathcal{B}$ is computably isomorphic to $\mathcal{A}$.

(ii) No low_2 members, which follows by work of Feiner [13].
(iii) A *single* element $\mathbf{0}'$, by Downey and Moses [11].
(iv) Every c.e. degree *except* $\mathbf{0}$, by Downey [7, Theorem 2.9], using work of Downey and Knight [10] and R. Miller [25].
(v) All c.e. degrees above any given nonzero c.e. degree, a result implicit in Downey [6, Theorem 7.5], using the method of Downey [6, Theorem 7.7], based on Downey and Moses [11], Downey and Knight [10], and Knight [22]; this result appears explicitly in Chubb, Frolov and Harizanov [4, Theorem 2.5], based on Downey and Moses [11].

These results use coding methods and *separator* results, building on work of Jockusch and Soare [21]. A much more subtle question asks what can be said about the spectrum of the successivity relation of a *given* linear ordering. We have already seen that if the ordering has a copy where $\text{Succ}(\mathcal{L})$ is computable then the spectrum of $\text{Succ}(\mathcal{L})$ includes all c.e. degrees, but that there are examples where $\text{Succ}(\mathcal{L})$ is not computable in any computable copy of $\mathcal{L}$.

The goal of this paper is to solve a fairly long-standing question (essentially going back to [11]), which asks whether there is a computable linear ordering with infinitely many successivities where the successivity relation is *intrinsically incomplete* (in the sense of Ash and Nerode [3]). That is, does there exist a computable linear ordering $\mathcal{L}$ with infinitely many successivities such that $\mathbf{0}'$ is not in the spectrum of $\text{Succ}(\mathcal{L})$? It was known by work of Downey and Moses [11] that the answer is yes for the *weak truth-table* degrees, and that the answer is no for *discrete* linear orderings (namely, ones where every element save possibly the first and the last (if any) has a successor and a predecessor), but the question has resisted all attempts at a solution in the general case. In the present paper, we solve this question negatively by showing that it is always possible to code $\mathbf{0}'$ into the spectrum of $\text{Succ}(\mathcal{L})$ for any computable linear ordering $\mathcal{L}$ with infinitely many successivities.

Main Theorem. *Let $\mathcal{A}$ be an infinite computable linear ordering with infinitely many successivities. Suppose that C is any c.e. set with $\text{Succ}(\mathcal{A}) \leq_T C$. Then there is a computable linear ordering $\mathcal{B}$ isomorphic to $\mathcal{A}$ whose successivity relation has Turing degree $\deg_T(C)$.*

Although we believe that the above theorem can be proved uniformly, the proof seems far easier when relying on the following partial solution to this problem by Frolov:

Theorem 1.2 (Frolov [14]). *Let $\mathcal{A}$ be an infinite computable linear ordering with infinitely many successivities which is either not η-like or which is strongly η-like. Suppose that C is any c.e. set with $\mathrm{Succ}(\mathcal{A}) \leq_T C$. Then there is a computable linear ordering $\mathcal{B}$ isomorphic to $\mathcal{A}$ whose successivity relation has Turing degree $\deg_T(C)$.*

Here, a linear ordering is *η-like* if it contains no interval of order-type ω or ω^*; and it is *strongly η-like* if in addition there is a fixed finite bound on the size of the finite intervals. (We use the word "interval" here simply to denote a "convex" subset, i.e., an interval need not have endpoints in the linear ordering.) Note that Frolov's result immediately extends to infinite linear orders which contain an infinite strongly η-like interval with infinitely many successivities.

Frolov's result can also be extracted from Chubb, Frolov and Harizanov [4], who proved the following, which can then be applied to the appropriate interval of $\mathcal{L}$.

Theorem 1.3 (Chubb, Frolov and Harizanov [4]). *If $\mathcal{L}$ is a computable linear ordering where for all $x \in L$ there is a successivity (a, b) in $\mathcal{L}$ with $x <_{\mathcal{L}} a$, then the spectrum of the successivity relation of $\mathcal{L}$ is closed upwards in the c.e. Turing degrees.*

The remainder of this paper is devoted to the proof of our Main Theorem in the case not handled by Frolov's theorem. As we remarked in the abstract, the proof has some rather unusual features in that we do not construct partial isomorphisms in the construction but certain partial 1–1 maps, and that the final isomorphism can only be read off from the true path of the construction, and even along it only in the limit, making our isomorphism only Δ^0_3. The point is that there certainly have been Δ^0_3-isomorphisms in the past, such as Downey [5], but even there, partial bijections were kept in a stage-by-stage construction. This is not true here. Our method here has already found other applications such as Downey, Kastermans and Lempp [9], where a Δ^0_3-isomorphism is constructed similarly to our method here, and we believe it promises many more. In retrospect, it is clear that this feature of the construction is *precisely* what is needed to overcome the problems caused by the failure of the hypothesis of Theorem 1.3.

We refer to Ash/Knight [2] for basic notions from computable algebra and computable model theory, and to Soare [27] for basic notation from computability theory. In particular, recall that for nodes σ, σ' on a tree T, $\sigma <_L \sigma'$ denotes that for some n, $\sigma(n) < \sigma'(n)$ and $\sigma \upharpoonright n = \sigma' \upharpoonright n$; and that $\sigma < \sigma'$ denotes that $\sigma <_L \sigma'$ or $\sigma \subset \sigma'$.

2. Intuition for the Proof of the Main Theorem

For simplicity of presentation, we will prove the main result for $C = \emptyset'$ and make some remarks at the end on how to improve this to upper cones.

To fix terminology, in an η-like linear ordering $\mathcal{A}$, we will call the maximal finite interval containing an element $a \in A$ the *maximal block* of a. More generally, any finite interval in $\mathcal{A}$ will be called a *block*.

As mentioned above, the key idea of our proof is similar to the (somewhat easier) proof technique in Downey, Kastermans, Lempp [9], who proved that for any η-like computable linear ordering without infinite strongly η-like interval, there is a computable isomorphic copy without nontrivial computable self-embedding. The idea there was not to try to produce a Δ^0_2-isomorphism by effectively approximating it by finite partial isomorphisms, but to define finite parts of a Δ^0_3-isomorphism along the true path of an infinite-injury priority argument on a tree of strategies. Adapting this idea to our setup, each strategy on the tree of strategies working on making the isomorphism ι total and onto tries to map one more maximal block of elements in $\mathcal{A}$ to a maximal block in $\mathcal{B}$ of corresponding size; however, the present argument introduces additional difficulties we will address below.

We will thus fix an infinite η-like computable linear ordering $\mathcal{A} = (A, <_{\mathcal{A}})$ with infinitely many successivities. We need to build a computable linear ordering $\mathcal{B} = (B, <_{\mathcal{B}})$ isomorphic to $\mathcal{A}$ and a (non-computable) map $\iota : A \to B$, meeting, in increasing order of difficulty, the following

Overall Requirements:

- $\mathcal{O}$: ι is order-preserving, i.e., for all $a, a' \in A$, $a <_{\mathcal{A}} a'$ implies $\iota(a) <_{\mathcal{B}} \iota(a')$ (and so in particular ι is injective);
- $\mathcal{W}$: ι is well-defined (and in particular total);
- $\mathcal{S}$: ι is surjective; and
- $\mathcal{R}$: there is a Turing reduction Γ such that $\Gamma^{\mathrm{Succ}(\mathcal{B})} = C$, where $\mathrm{Succ}(\mathcal{B})$ is the successivity relation in $\mathcal{B}$ and C is a 1-complete c.e. set.

Since $\mathcal{A}$ is η-like, we may without loss of generality (but non-uniformly) assume that $\mathcal{A}$ has neither a least nor a greatest element (since removing at most finitely many elements from an η-like linear ordering will always result in a linear ordering without endpoints). We will also (again non-uniformly) choose a successivity (a_0, a_1) in $\mathcal{A}$ and assume that these two elements are enumerated into $\mathcal{A}$ first. (It is possible to remove these non-uniformities, but our assumptions of no endpoints and of a fixed successivity make our construction easier.)

The isomorphism ι will now be defined along the true path TP, in the sense that ι is the limit of finite partial isomorphisms ι_σ between $\mathcal{A}$ and $\mathcal{B}$ for $\sigma \subset TP$. (Unlike in Downey, Kastermans and Lempp [9], however, while ι will still be the increasing union of the finite partial isomorphisms ι_σ for $\sigma \subset TP$, there will be times when we have to correct Γ, in which case we will have to "reset" our isomorphism back, destroying successivities in $\mathcal{B}$ and causing additional initializations along the true path. So while our isomorphism ι will again only be Δ^0_3, one can no longer compute the isomorphism ι from the true path and the function giving the stage beyond which the approximation to the true path no longer moves left of the true path up to level n of the tree of strategies, as was possible in [9].)

In the absence of the requirement for building $\Gamma^{\mathrm{Succ}(\mathcal{B})}$, it is, of course, trivial to build a computable isomorphic copy $\mathcal{B}$ of $\mathcal{A}$, by simply copying all of $\mathcal{A}$ into $\mathcal{B}$. However, we will present here a different way of organizing the construction of $\mathcal{B}$ and the isomorphism ι, with an eye toward later extending it to the full construction.

2.1. **Making ι order-preserving.** This is the simplest requirement: We simply ensure that we do not make the "silly" mistake of letting ι map elements in $\mathcal{A}$ to elements in $\mathcal{B}$ that are ordered differently.

The remaining overall requirements can be split up into the following *Requirements:*

$\mathcal{W}_a$: $\iota(a)$ is (well-)defined, for each $a \in A$;
$\mathcal{S}_b$: $\iota^{-1}(b)$ is defined, for each $b \in B$; and
$\mathcal{R}_i$: $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i) = C(i)$ for each $i \in \omega$.

2.2. **Making ι well-defined.** Our technique for meeting this requirement foreshadows the technique we introduce in section 2.5. We ensure that ι is well-defined not by ensuring it one element at a time, but one maximal block at a time, using the fact that $\mathcal{A}$ is η-like and so each maximal block is finite. So fix an element $a \in A$ for which $\iota(a)$ has not yet been defined. We need to guess the (finite) size of the maximal block in $\mathcal{A}$ containing a. Note that this can be done uniformly in a Π^0_2-fashion in the sense that there are two computable functions $LB, RB : A \times \omega \to \omega$ such that the size of the maximal block containing a to the left and right of a is given by $\liminf_s LB(a,s)$ and $\liminf_s RB(a,s)$, respectively.

Now, at stage s, we simply ensure that there are $LB(a,s) + 1 + RB(a,s)$ many elements in $\mathcal{B}$ to which the maximal block of a can be mapped. Whenever $LB(a,s)$ or $RB(a,s)$ increases (and so in particular when we first start working on the element a), we add more elements

to $\mathcal{B}$ immediately to the left or right of a, respectively, in order to have a maximal block in $\mathcal{B}$ of the correct size. Whenever $LB(a,s)$ or $RB(a,s)$ decreases, we discard some of the elements in $\mathcal{B}$ previously in the ι-image of the maximal block of a from the current range of ι; of course, these elements cannot be removed from $\mathcal{B}$, but they are now no longer in the range of (our new version of) ι, and the strategies making ι surjective will worry about supplying preimages for these elements of $\mathcal{B}$ later on.

2.3. **Making ι surjective.** Such a strategy σ will in general work within an interval of $\mathcal{B}$, possibly bordered on one or both sides by maximal blocks, which determine (under ι^{-1}) an infinite interval of $\mathcal{A}$, again possibly bordered on one or both sides by maximal blocks. Fix an element b in some infinite interval of $\mathcal{B}$ which is not currently in the range of ι. We now simply find an element a in our interval of $\mathcal{A}$, identify the maximal block containing a, and proceed as in section 2.2 to map a and its maximal block to $\mathcal{B}$.

2.4. **Defining Γ at an argument i.** The most complicated type of requirement tries to compute C from $\mathrm{Succ}(\mathcal{B})$ via Γ. A typical strategy σ for this requirement, trying to define Γ at an argument i, will define the use of the computation $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ to include the images in $\mathcal{B}$ of all successivities in all blocks in $\mathcal{A}$ found by strategies $\subset \sigma$. There will always be at least one such successivity since we fixed the successivity (a_0, a_1) in $\mathcal{A}$ beforehand and we assigned this successivity to the highest-priority strategy on the tree of strategies, which will be a $\mathcal{W}$-strategy defining $\iota(a_0)$ (and thus by section 2.2 also $\iota(a_1)$); so any computation $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ has nontrivial use and can be corrected by destroying some successivity in $\mathcal{B}$. Furthermore, eventually $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ will be defined with $\mathrm{Succ}(\mathcal{B})$-correct use since any strategy on the true path eventually identifies only correct blocks in $\mathcal{A}$, and thus in $\mathcal{B}$, at stages when it has the correct outcome.

When σ sees that $C(i)$ has changed and wants to correct $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$, σ will try to minimize the injury to higher-priority strategies: If there is a successivity in the use of $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ not contained in any block controlled by a higher-priority strategy, then σ will simply destroy such a successivity by inserting an element into $\mathcal{B}$. On the other hand, it may be that the use of $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ only contains successivities contained in blocks controlled by higher-priority strategies τ. In the style of minimizing injury as in the Sacks Splitting Theorem, σ will now identify a successivity contained only in maximal blocks of strategies τ of lowest possible priority. Given that $\mathcal{A}$, and thus also $\mathcal{B}$, contains infinitely

many (maximal) blocks of size > 1 and that more and more such blocks will be identified along the true path, and given that each computation $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ needs to be corrected at most once (since C is c.e. and the Γ-oracle $\mathrm{Succ}(\mathcal{B})$ is co-c.e.), the injury to each maximal block of $\mathcal{B}$ found along the true path will be finite.

2.5. **Identifying maximal blocks and outcomes of strategies.** We now introduce the key ideas of how we combine the strategies. We first discuss the outcomes of the $\mathcal{W}_a$- and $\mathcal{S}_b$-strategies. (Since each $\mathcal{R}_i$-strategy acts at most once (in a way which affects other strategies, namely, when i enters C), we allow only one outcome for an $\mathcal{R}_i$-strategy.)

The main problem consists in defining the map ι while trying to keep fixed (finite) blocks in $\mathcal{B}$ intact as much as possible (i.e., not inserting additional elements into them unless required for Γ-correction) in order to have them serve as images of blocks in $\mathcal{A}$.

The setup will consist of a strategy σ being given a finite collection of finite blocks in $\mathcal{A}$ and their ι-images in $\mathcal{B}$ identified by strategies $\subset \sigma$. Suppose σ tries to define $\iota(a)$ or $\iota^{-1}(b)$. In the latter case, we first identify the element a with least Gödel number in the interval between the ι-preimages of the $\mathcal{B}$-blocks between which b lies. (If there is no such a, then we end the stage; this delay must be finite if σ is along the true path.) Next, we need to identify a maximal block $\overline{a}'$ in $\mathcal{A}$ of length $m+1+n$, say, containing a as well as m and n many points to the left and right of a, respectively. Then we let ι map $\overline{a}'$ to a block $\overline{b}'$ (containing b if we are defining $\iota^{-1}(b)$). We will "guess" the block $\overline{a}'$ in $\mathcal{A}$ by first guessing the length $m+1+n$ of the maximal block in $\mathcal{A}$ containing a, associating with the guess (m,n) each time an $(m+1+n)$-tuple $\overline{a}'$ of elements in $\mathcal{A}$ which we currently believe to form a maximal block in $\mathcal{A}$ containing a.

The *outcomes* of a strategy trying to define $\iota(a)$ or $\iota^{-1}(b)$ are thus of the form $(m,n,\overline{a}')$, where $m,n \in \omega$ and $\overline{a}'$ is a $<_{\mathcal{A}}$-ordered $(m+1+n)$-tuple of consecutive elements from $\mathcal{A}$ containing a. These outcomes are ordered lexicographically, where the ordering on $\overline{a}'$ is by Gödel number.

We can organize the guessing for a "correct" triple $(m,n,\overline{a}')$ (i.e., a triple such that $\overline{a}'$ is a maximal block in $\mathcal{A}$) such that this triple is the leftmost outcome guessed infinitely often. In addition, there will be a stage s' such that at any later stage, the only guesses on the outcome will be the "correct" triple $(m,n,\overline{a}')$ and triples of the form $(m',n',\overline{a}'')$ (where $m' > m$, or $m' = m$ and $n' > n$). (This is because once we have correctly guessed this $\overline{a}'$ for the first time, we will from then on only

be wrong by guessing tuples $\overline{a}''$ which are strictly longer to the left or right of a.)

2.6. **Interaction between incomparable strategies working for the same requirement.** In order for the construction overall to succeed, we need to be very careful about the interaction between incomparable strategies working for the same requirement. This is obvious for an $\mathcal{R}_i$-requirement: All $\mathcal{R}_i$-strategies together define the functional Γ at a single argument i; so any definition of $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ made by one $\mathcal{R}_i$-strategy has to be dealt with by all other $\mathcal{R}_i$-strategies as well, even if the final definition of $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ is made by an $\mathcal{R}_i$-strategy off the true path. However, this will not be as difficult at it appears at first, since we only have to ensure that $\Gamma^{\mathrm{Succ}(\mathcal{B})}$ is total, and that we can correct $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ when i enters C.

The interaction between incomparable strategies for the other requirements will turn out to be much more subtle, as we will now explain. Indeed, we will need to ensure the following property of the construction to hold: If σ and σ' are two strategies for the same $\mathcal{W}_a$- or $\mathcal{S}_b$-requirement, respectively, with $\sigma <_L \sigma'$, then one of the following two cases must hold at any stage (where $a = \iota^{-1}(b)$, as defined by σ' or some $\tau \subset \sigma'$, in the case of an $\mathcal{S}_b$-requirement):

(1) Either $\iota(a)$ is defined by a strategy $\tau \subset \sigma'$ (as part of a maximal block found by τ, which happens to include a); or
(2) the maximal block $\overline{a}'$ found by σ in defining $\iota(a)$ is a proper subblock of the maximal block $\overline{a}''$ found by σ', and σ and σ' agree on the value of $\iota(a)$.

We need the above property in order to avoid the following problem: If σ is an $\mathcal{S}_b$-strategy to the left of the true path trying to define $a = \iota^{-1}(b)$, then σ may believe that a is part of a large maximal block $\overline{a}'$. But if σ and σ' try to define $\iota^{-1}(b)$ differently, then σ' may believe that $a' = \iota^{-1}(b)$ is part of a smaller maximal block and so will try to make the maximal block around b smaller than allowed by σ, which would result in injury by the lower-priority strategy σ' to the higher-priority strategy σ and therefore cannot be allowed since after all, σ may be on the true path and then cannot afford to be injured by lower-priority strategies infinitely often.

Furthermore, note that we can indeed ensure the above property, proceeding by induction on the length of σ and σ' (which we may assume to be equal by the assignment of requirements to nodes on the tree of strategies in section 3.1). Assume that clause (1) does not apply to σ and σ' and that the property holds for all shorter nodes. Then, by induction on $|\sigma|$, both σ and σ' work with the same maximal blocks

provided by the strategies $\tau \subset \sigma$ and $\tau' \subset \sigma'$, respectively, except that the blocks provided by the strategies τ' may properly contain blocks provided by the strategies τ, and that some of the blocks provided by the strategies τ may be combined into single blocks provided by the strategies τ'. However, by the failure of clause (1) for σ', and by clause (2) for shorter strategies, we can assume that a lies between σ-subblocks of σ'-blocks (in the case of a $\mathcal{W}_a$-requirement), or that b lies between the ι-images of σ-subblocks of σ'-blocks (in the case of an $\mathcal{S}_b$-requirement, respectively). This allows σ' to define $\iota(a)$ respecting clause (2).

In ensuring (2), there is one obstacle we need to overcome with a small trick, under the notation of the previous paragraph: Normally, one would leave any restraint imposed by σ intact while σ appears to be to the left of the true path. However, for the same reason as mentioned two paragraphs above, we cannot afford to do so: It may turn out that σ is wrong about a large block around a and so would want to protect a large block around $b = \iota(a)$ even though while σ is to the left of the current true path, it turns out that σ's guess about the size of the maximal block containing a was too high. The solution to this problem is now obvious: Even though σ is to the left of the current true path, σ will continue to monitor its guess about the size of the block containing a, but it will only decrease its guess as elements enter near a to the left and right of a. This trick will ensure that whatever guess σ currently has about the size of the maximal block containing a, it is no larger than the corresponding guess of σ', and so σ will not try to protect a larger block around $b = \iota(a)$ than σ'.

3. Proof of the Main Theorem

In the following, we will assume familiarity with priority arguments on a tree of strategies (cf., e.g., Soare [27] or Lempp [23]).

3.1. **The tree of strategies.** The full construction takes place on a *tree of strategies* $T = \Lambda^{<\omega}$ where

$$\Lambda = \{(m, n, \overline{a}') \mid m, n \in \omega \text{ and } \overline{a}' \in A^{m+1+n}\} \cup \{\mathit{fin}\}$$

is the *set of outcomes* of a strategy, ordered lexicographically, with fin as the greatest element, and where the $\overline{a}'$ are ordered by Gödel numbers.

We effectively order all requirements in a list $\{\mathcal{Q}_j\}_{j\in\omega}$ such that $\mathcal{W}_{a_i} = \mathcal{Q}_{3i}$, $\mathcal{S}_{b_i} = \mathcal{Q}_{3i+1}$ and $\mathcal{R}_i = \mathcal{Q}_{3i+2}$ for all $i \in \omega$ where $A = \{a_i \mid i \in \omega\}$ and $B = \{b_i \mid i \in \omega\}$, and where

$\mathcal{W}_a$: $\iota(a)$ is (well-)defined,
$\mathcal{S}_b$: $\iota^{-1}(b)$ is defined, and

$\mathcal{R}_i$: $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i) = C(i)$.

We will assume that the highest-priority requirement $\mathcal{Q}_0$ is the $\mathcal{W}_{a_0}$-requirement so that the successivity (a_0, a_1) is found immediately and can be included in all Γ-uses. Furthermore, all strategies $\sigma \in T$ of length j are assigned to requirement $\mathcal{Q}_j$.

3.2. **The full strategies.** Each strategy is equipped with a finite partial map $\iota_\sigma : A \to B$, which is its current guess about the isomorphism $\iota : \mathcal{A} \to \mathcal{B}$. Naturally, we will want $\tau \subset \sigma \in T$ to imply $\iota_\tau \subseteq \iota_\sigma$; so a strategy σ will always have to live with

$$\iota_\sigma^- := \bigcup_{\tau \subset \sigma} \iota_\tau$$

Furthermore, we define, for each strategy σ, the *set S_σ of stages at which σ is eligible to act* by

$$S_\sigma = \{s \mid \sigma \subseteq TP_s\}$$

where TP_s is our approximation to the true path of the construction at stage s (to be defined in section 3.3).

Finally, each $\mathcal{W}$- and $\mathcal{S}$-strategy σ is also associated with two "block size" functions, approximated by two computable functions (denoted, with some abuse of notation, by the same letters)

$$LB_\sigma, RB_\sigma : A \times \omega \to \omega,$$

which, for all such σ along the true path TP, will have the property that

$$\begin{aligned} LB_\sigma(a) &:= \liminf_{s \in S_\sigma} LB_\sigma(a, s) \\ &= \text{size of maximal block in } \mathcal{A} \text{ containing } a \text{ left of } a \\ RB_\sigma(a) &:= \liminf_{s \in S_\sigma} RB_\sigma(a, s) \\ &= \text{size of maximal block in } \mathcal{A} \text{ containing } a \text{ right of } a \end{aligned} \tag{1}$$

Specifically, we define two computable functions, for any stage s and for the greatest $s' \in S_\sigma$ with $s' < s$, as follows:

$LB_\sigma(a, s) = |[a', a]| - 1$, where a' is $<_{\mathcal{A}}$-least such that no element has entered $\mathcal{A}$ in the interval $[a', a]$ since stage s', and

$RB_\sigma(a, s) = |[a, a']| - 1$, where a' is $<_{\mathcal{A}}$-greatest such that no element has entered $\mathcal{A}$ in the interval $[a, a']$ since stage s'.

Intuitively, the functions LB_σ and RB_σ guess at the size of the part of the maximal block containing a to the left and to the right of a, respectively. Clearly, these two functions will have liminf equal to the true sizes of the part of the block to the left and right of a, respectively.

It is now easy to check that this definition of the functions LB_σ and RB_σ ensures (1) above for all $\mathcal{W}$- and $\mathcal{S}$-strategies σ along the true path (and indeed for all strategies σ for which S_σ is infinite). Note furthermore that we define the block size function not only for stages $s \in S_\sigma$ but for all stages; this is because we need to measure how the guess on the block size to the left and right of a decreases for strategies σ to the left of the current true path as explained in section 2.6.

In the following, we will describe the action of a strategy $\sigma \in T$ depending on the type of requirement it is assigned to. We describe the action of each type of strategy at a stage s and let s' be the previous stage in S_σ since σ's most recent initialization. (We set $s' = s$ if no such stage exists.)

3.2.1. *The full $\mathcal{W}_a$-strategy.* This strategy σ has to ensure that $\iota(a)$ is defined. If $\iota_\sigma^-(a)$ is already defined, then the strategy simply ends the substage with outcome *fin*.

Otherwise, the strategy guesses that the maximal block containing a also contains the $LB_\sigma(a,s)$ many elements currently immediately to the left of a as well as the $RB_\sigma(a,s)$ many elements currently immediately to the right of a; we will denote this tuple of elements in $\mathcal{A}$ by $\overline{a}'$. The outcome of the strategy is now $(LB_\sigma(a,s), RB_\sigma(a,s), \overline{a}')$ (denoting that the strategy guesses that the maximal block containing a consists of the $LB_\sigma(a,s) + 1 + RB_\sigma(a,s)$ many elements in $\overline{a}'$). Then the strategy defines $\iota_\sigma(\overline{a}')$ as follows:

- If $s' = s$ (i.e., if this is the first stage at which σ is eligible to act since its most recent initialization), then check whether there is a (lowest-priority) $\mathcal{W}_a$-strategy $\sigma_0 <_L \sigma$ which has last been eligible to act at a stage $s_0 < s$, say, and has not been initialized since then. If so, then set $L = LB_{\sigma_0}(a,s)$, $R = RB_{\sigma_0}(a,s)$, and define $\iota_\sigma(a) = \iota_{\sigma_0}(a)$; otherwise, set $L = R = 0$ and define $\iota_\sigma(a)$ to be the element $b \in B$ with least Gödel number consistent with ι_σ^- (if no such element b currently exists in $\mathcal{B}$, then create one). Now create L many new elements in $\mathcal{B}$ immediately to the left of b, and R many new elements immediately to the right of b; call these $L+1+R$ many elements $\overline{b}'$, and define $\iota_\sigma(\overline{a}') = \overline{b}'$.
- Otherwise, first check whether $LB_\sigma(a,s') < LB_\sigma(a,s)$. If so, then create $LB_\sigma(a,s) - LB_\sigma(a,s')$ many new elements immediately to the left of the current $LB_\sigma(a,s')$ many elements immediately to the left of $b = \iota_\sigma(a)$. Next check whether $RB_\sigma(a,s') < RB_\sigma(a,s)$. If so, then create $RB_\sigma(a,s) - RB_\sigma(a,s')$ many new elements immediately to the right of the current $RB_\sigma(a,s')$ many

elements immediately to the right of b. Finally, denote by $\overline{b}'$ the tuple in $\mathcal{B}$ consisting of the $LB_\sigma(a,s)$ many elements now immediately to the left of b, b itself, as well the $RB_\sigma(a,s)$ many elements now immediately to the right of b. Define $\iota_\sigma(\overline{a}') = \overline{b}'$.

The strategy now ends the substage with outcome

$$(LB_\sigma(a,s), RB_\sigma(a,s), \overline{a}').$$

3.2.2. *The full $\mathcal{S}_b$-strategy.* This strategy σ has to ensure that $\iota^{-1}(b)$ is defined. If $(\iota_\sigma^-)^{-1}(b)$ is already defined, then the strategy simply ends the substage with outcome *fin*.

Otherwise, the strategy defines $\iota_\sigma^{-1}(b)$ as follows:

- If $s' = s$ (i.e., if this is the first stage at which σ is eligible to act since its most recent initialization), then check whether there is a (lowest-priority) $\mathcal{S}_b$-strategy $\sigma_0 <_L \sigma$ which has last been eligible to act at a stage $s_0 < s$, say, and has not been initialized since then. If so, then define $a = \iota_\sigma^{-1}(b) = \iota_{\sigma_0}^{-1}(b)$ and set $L = LB_{\sigma_0}(a,s)$ and $R = RB_{\sigma_0}(a,s)$; otherwise, set $L = R = 0$ and define $\iota_\sigma^{-1}(b)$ to be the element $a \in A$ with least Gödel number consistent with ι_σ^- (if no such element a currently exists in $\mathcal{A}$, then end the stage; this wait must be finite if σ is on the true path). Now create L many new elements in $\mathcal{B}$ immediately to the left of b, and R many new elements immediately to the right of b; call these $L + 1 + R$ many elements $\overline{b}'$, and define $\iota_\sigma(\overline{a}') = \overline{b}'$.
- Otherwise, let $a = \iota_\sigma^{-1}(b)$ and first check whether $LB_\sigma(a,s') < LB_\sigma(a,s)$. If so, then create $LB_\sigma(a,s) - LB_\sigma(a,s')$ many new elements immediately to the left of the current $LB_\sigma(a,s')$ many elements immediately to the left of $b = \iota_\sigma(a)$. Next check whether $RB_\sigma(a,s') < RB_\sigma(a,s)$. If so, then create $RB_\sigma(a,s) - RB_\sigma(a,s')$ many new elements immediately to the right of the current $RB_\sigma(a,s')$ many elements immediately to the right of b. Finally, denote by $\overline{b}'$ the tuple in $\mathcal{B}$ consisting of the $LB_\sigma(a,s)$ many elements now immediately to the left of b, b itself, as well the $RB_\sigma(a,s)$ many elements now immediately to the right of b. Define $\iota_\sigma(\overline{a}') = \overline{b}'$.

The strategy now ends the substage with outcome

$$(LB_\sigma(a,s), RB_\sigma(a,s), \overline{a}').$$

3.2.3. *The full $\mathcal{R}_i$-strategy.* This strategy needs to ensure $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i) = C(i)$.

We distinguish four cases for the action of this strategy σ.

If there is some $\mathcal{S}$- or $\mathcal{W}$-strategy $\tau \subset \sigma$ which has not been eligible to act before stage s since its most recent initialization, then σ ends the stage. (This is to ensure that the block size measurements of such τ make sense.)

Otherwise, if $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ is currently defined but not equal to $C(i)$, then σ will make $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ undefined as follows. (Note that there may be several definitions of $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ applying, in which case the below procedure has to be performed for each of them.)

- If the use $\gamma(i)$ includes a successivity which is not in a block in $\mathcal{B}$ in the range of ι_σ^-, then σ simply inserts a new element into $\mathcal{B}$ so as to destroy the leftmost such successivity; in addition, σ initializes all $\tau' > \sigma$ and ends the stage.
- Otherwise, fix the shortest $\tau \subset \sigma$ such that the use $\gamma(i)$ includes only successivities which are in a block in $\mathcal{B}$ in the range of ι_τ. Then σ inserts a new element into $\mathcal{B}$ so as to destroy the leftmost successivity in $\gamma(i)$ which is not in a block in $\mathcal{B}$ in the range of ι_τ^-; in addition, σ initializes all strategies $\tau' \geq \tau$ and ends the stage. (Note that this will in particular imply that σ initializes itself.)

Otherwise, and if $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ is not currently defined by a definition made by σ, then σ defines $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i) = C(i)$ with use $\gamma(i)$ including all the successivities in all blocks in the range of ι_σ^-. (Note that there will always be at least one such successivity, namely, (a_0, a_1).) In addition, σ initializes all strategies $\tau > \sigma$ and ends the stage.

Finally, if neither of the above cases applies, then σ takes no action but simply takes outcome *fin*.

3.3. **The construction.** The construction proceeds in stages $s \in \omega$, each subdivided into substages $t \leq s$.

At the beginning of stage 0, we initialize all strategies in T.

Stage s, substage t: At substage t of stage s, the strategy σ of length t with the currently correct guess about the outcomes of the strategies $\tau \subset \sigma$ is *eligible to act*. At the end of each substage $t \leq s$, the strategy σ will either end the stage (if its description specifies this, or if $s = t$), or determine its outcome o and thus the strategy $\sigma^\frown\langle o\rangle$ eligible to act at the next substage $t+1$. The action of each strategy is determined by the type of requirement it is assigned to and is as described in section 3.2.

Let TP_s be the longest strategy eligible to act at stage s. At the end of each stage, any strategy $\sigma >_L TP_s$ is initialized.

This completes the description of the construction.

3.4. **The verification.** We first establish a lemma which will help us show that each strategy along the true path cannot be initialized infinitely often by Γ-correction:

Lemma 3.1. *Let C_0 be the set of $i \in C$ such that at some stage, a (highest-priority) strategy τ_i, say, is initialized when an $\mathcal{R}_i$-strategy corrects $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ by adding one or more elements into $\mathcal{B}$. Then for each $i \in C_0$, there is only one such τ_i, and*

$$\lim_{i \in C_0} |\tau_i| = \infty.$$

Proof. Fix n arbitrary. Using η-likeness and the fact that $\mathcal{A}$ contains infinitely many successivities, fix $m > n$ such that the first m many elements of $\mathcal{A}$ are contained in maximal blocks of size > 1. Next, fix a stage s_0 such that by stage s_0, the finite maximal blocks of the first m many elements of $\mathcal{A}$ as well as $C \upharpoonright m$ have been completely enumerated. Since the set of possible outcomes of each strategy is well-ordered (of order-type $\omega^3 + 1$), we can fix a stage $s_1 \geq s_0$ such that at all stages $s > s_1$, $TP_s \upharpoonright 3m \geq TP_{s_1} \upharpoonright 3m$. Finally, fix a stage $s_2 \geq s_1$ such that no computation $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ defined by stage s_1 is corrected after stage s_2.

We now claim that after stage s_2, $|\tau_i| \geq 3n$ for all $i \in C_0$ for which $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ is corrected after stage s_2. This is because any such $\mathcal{R}_i$-strategy σ, say, correcting $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ must have length $> 3m$ and be preceded by $> n$ many $\mathcal{W}$-strategies having found blocks (possibly properly) containing maximal blocks of size > 1. Thus τ_i must be longer than the nth $\mathcal{W}$-strategy, giving $|\tau_i| \geq 3n$ as desired. □

We are now ready to prove the strategies along the true path ensure the satisfaction of all requirements, finishing up the proof of our Main Theorem for the case $C = \emptyset'$:

Lemma 3.2. *There is an infinite path TP through the tree T of strategies (called the* true path*) with the following properties for all $j \in \omega$:*

- *(1) $TP \upharpoonright j$ is eligible to act infinitely often.*
- *(2) Any $\sigma <_L TP \upharpoonright j$ is eligible to act at most finitely often.*
- *(3) $TP \upharpoonright j$ is initialized at most finitely often.*
- *(4) If $\sigma = TP \upharpoonright j$ is a $\mathcal{W}_a$-strategy, then either $\iota_\sigma^-(a)$ is already defined, or there is a leftmost outcome o taken infinitely often by σ, and at almost all stages at which σ has outcome o,*
  - *(a) σ correctly identifies the maximal block in $\mathcal{A}$ containing a, and*
  - *(b) $\iota_\sigma(\overline{a}')$ takes a fixed value for the maximal block $\overline{a}'$ containing a.*

(5) *If $\sigma = TP \upharpoonright j$ is an $\mathcal{S}_b$-strategy, then either $(\iota_\sigma^-)^{-1}(b)$ is already defined, or there is a leftmost outcome o taken infinitely often by σ, and at almost all stages at which σ has outcome o,*
   (a) *σ defines $a = \iota(b)$ for some fixed a in $\mathcal{A}$,*
   (b) *σ correctly identifies the maximal block $\overline{a}'$ in $\mathcal{A}$ containing a, and*
   (c) *$\iota_\sigma(\overline{a}')$ takes a fixed value for the maximal block containing a.*

(6) *If $\sigma = TP \upharpoonright j$ is an $\mathcal{R}_i$-strategy then $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ is defined and correctly computes $C(i)$.*

Proof. We proceed by induction on j and assume the lemma for all $j' < j$.

Clauses (1) and (2) are trivial for $j = 0$, and for $j > 0$, they follow by the facts that

- the set of possible outcomes of $TP \upharpoonright (j-1)$ is well-ordered (of order-type $\omega^3 + 1$),
- the delay in case $TP \upharpoonright (j-1)$ is an $\mathcal{S}_b$-strategy waiting for an ι-preimage for b must be bounded, and
- $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ can be corrected at most once for each i.

Clause (3) now follows easily by Lemma 3.1.

We complete the proof of our lemma by distinguishing cases for the type of requirement assigned to $\sigma = TP \upharpoonright j$.

Suppose first that σ is a $\mathcal{W}_a$-strategy and that $\iota_\sigma^-(a)$ is not defined. Then σ eventually takes as its outcome only the true outcome $o = (m, n, \overline{a}')$ for the correct values m, n and $\overline{a}'$ (for the size m of the maximal block to the left of a, the size n of the maximal block to the right of a, and the maximal block $\overline{a}'$ containing a) as well as outcomes of the type $(m', n', \overline{a}'')$ (for $m' > m$) or $(m, n', \overline{a}'')$ (for $n' > n$), so the true outcome o satisfies clause (4) and σ correctly identifies the maximal block containing a whenever σ takes outcome o. Finally, since the definitions of $\iota_\sigma(a)$ and $\iota_{\sigma_0}(a)$ agree for all $\mathcal{W}_a$-strategies $\sigma_0 < \sigma$ for which $\iota_{\sigma_0}(a)$ is defined, and since $LB_\sigma(a, s) \geq LB_{\sigma_0}(a, s)$ and $RB_\sigma(a, s) \geq RB_{\sigma_0}(a, s)$ for all $\mathcal{W}_a$-strategies $\sigma_0 < \sigma$, σ eventually defines $\iota_\sigma(\overline{a}')$ the same way at all stages at which σ takes outcome o.

The argument for an $\mathcal{S}_b$-strategy σ is similar, except that we have to argue first that $\iota_\sigma^{-1}(b)$ stabilizes.

Finally, assume that σ is an $\mathcal{R}_i$-strategy. Clearly, by the construction, $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ cannot be permanently defined $\neq C(i)$, so we just need to show that $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$ is eventually defined with $\mathrm{Succ}(\mathcal{B})$-correct use. So suppose, for the sake of a contradiction, that no $\mathcal{R}_i$-strategy makes a permanent definition of $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$. Fix a stage s_0 after which σ is no

longer initialized and $C(i)$ no longer changes. Then σ will eventually make a $\Gamma^{\mathrm{Succ}(\mathcal{B})}(i)$-definition using only the successivities in the correct maximal blocks found by $\mathcal{W}$- and $\mathcal{S}$-strategies $\tau \subset \sigma$. □

In order to prove the full theorem, for an arbitrary c.e. degree $\geq \deg_T(\mathrm{Succ}(\mathcal{A}))$, the above construction simply needs to be combined with permitting and coding. So suppose we have a c.e. set $C \geq_T \mathrm{Succ}(\mathcal{A})$. There are only two reasons for us to destroy successivities in the copy $\mathcal{B}$ of $\mathcal{A}$ we are building. One is that the corresponding pair of elements in $\mathcal{A}$ is not really a successivity (which C can compute as it can compute $\mathrm{Succ}(\mathcal{A})$); and the other is for coding via Γ, which C can clearly compute as well. Thus the argument combines with permitting and coding in a standard way, since C can unravel the full construction.

References

[1] Ash, Christopher J., *A construction for recursive linear orderings*, J. Symbolic Logic **56** (1991), 673–683.

[2] Ash, Christopher J.; and Knight, Julia F., *Computable Structures and Hyperarithmetical Hierarchy*, Elsevier, 2000.

[3] Ash, Christopher J.; and Nerode, Anil, *Intrinsically recursive relations*, in: Aspects of Effective Algebra (Proc. Conf. Monash Univ., Clayton, Australia, Aug. 1-4, 1979), J. N. Crossley (ed.), 26–41.

[4] Chubb, Jennifer C.; Frolov, Andrey N.; and Harizanov, Valentina S., *Degree spectra of the successor relation of computable linear orderings*, Arch. Math. Logic **48** (2009), 7–13.

[5] Downey, Rodney G., *Every recursive boolean algebra is isomorphic to one with incomplete atoms*, Ann. Pure Appl. Logic **60** (1990), 193–206.

[6] Downey, Rodney G., *Computability Theory and Linear Orderings*, in: Handbook of Recursive Mathematics (eds. Ershov, Yuri L.; Goncharov, Sergey S.; Nerode, Anil; and Remmel, Jeffrey B.), Vol. 2, 823–976, North Holland, 1998.

[7] Downey, Rodney G., *Computability, definability and algebraic structures*, in: "Proceedings of the 7th and 8th Asian Logic Conferences", Singapore University Press, Singapore, 2003, 63–102.

[8] Downey, Rodney G; Goncharov, Sergey S.; and Hirschfeldt, Denis R., *Degree Spectra of relations on boolean algebras*, Algebra i Logika **42** (2003), no. 2, 105–111.

[9] Downey, Rodney G.; Kastermans, Bart; and Lempp, Steffen, *On computable self-embeddings of computable linear orderings*, J. Symbolic Logic **74** (2009), 1352–1366.

[10] Downey, Rodney G.; and Knight, Julia F., *Orderings with α-th jump degree $\mathbf{0}^\alpha$*, Proc. Amer. Math. Soc. **114** (1992), 545–552.

[11] Downey, Rodney G.; and Moses, Michael F., *Recursive linear orders with incomplete successivities*, Trans. Amer. Math. Soc. **326** (1991), 653–668.

[12] Dzgoev, Valeri D.; and Goncharov, Sergey S., *Autostability of models*, Algebra i Logika **19** (1980), no. 1, 45–58, 132.

[13] Feiner, L. J., *Orderings and Boolean Algebras not isomorphic to recursive ones*, Thesis, MIT (1967).
[14] Frolov, Andrey N., *Presentations of the adjacency relation of a computable linear order*, Izv. Vyssh. Uchebn. Zaved. Mat., to appear.
[15] Goncharov, Sergey S., *The number of nonautoequivalent constructivizations*, Algebra i Logika **16** (1977), no. 3, 257–282, 377.
[16] Harizanov, Valentina S., *Some effects of Ash–Nerode and other decidability conditions on degree spectra*, Ann. Pure Appl. Logic. **55** (1991), 51–65.
[17] Harizanov, Valentina S., *Uncountable degree spectra*, Ann. Pure Appl. Logic. **54** (1991), 255–263.
[18] Harizanov, Valentina S., *The possible Turing degree of the non-zero member in a two element degree spectrum*, Ann. Pure Appl. Logic **60** (1993), 1–30.
[19] Hirschfeldt, Denis R., *Degree spectra of relations on computable structures*, Bull. Symbolic Logic **6** (2000), 197–212.
[20] Hirschfeldt, Denis R.; Khoussainov, Bakhadyr M.; Shore, Richard A.; and Slinko, Arkadii M., *Degree spectra and computable dimensions in algebraic structures*, Annals of Pure and Applied Logic **115** (2002), 71–113.
[21] Jockusch, Carl G., Jr.; and Soare, Robert I., *Degrees of orderings not isomorphic to recursive linear orderings*, Ann. Pure Appl. Logic **52** (1991), 39–64.
[22] Knight, Julia F., *Degrees coded in jumps of orderings*, J. Symbolic Logic **51** (1986), 1034–1042.
[23] Lempp, Steffen, *Lecture Notes on Priority Arguments*, preprint available at `http://www.math.wisc.edu/~lempp/papers/prio.pdf`.
[24] Moses, Michael F., *Relations intrinsically recursive in linear orderings*, Z. Math. Logik Grundlag. Math. **32** (1986), 467–472.
[25] Miller, Russell G., *The Δ^0_2-spectrum of a linear order*, J. Symbolic Logic **66** (2001), 470–486.
[26] Remmel, Jeffrey B., *Recursively categorical linear orderings*, Proc. Amer. Math. Soc. **83** (1981), 387–391.
[27] Soare, Robert I., *Recursively enumerable sets and degrees*, Springer-Verlag, Berlin, New York, 1987.

School of Mathematics, Statistics and Operations Research, Victoria University of Wellington, Wellington, New Zealand
E-mail address: `Rod.Downey@mcs.vuw.ac.nz`
URL: `http://www.mcs.vuw.ac.nz/Main/RodDowney`

Department of Mathematics, University of Wisconsin, Madison, Wisconsin 53706-1388, USA
E-mail address: `lempp@math.wisc.edu`
URL: `http://www.math.wisc.edu/~lempp`

School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371, Republic of Singapore
E-mail address: `guohua@ntu.edu.sg`
URL: `http://www3.ntu.edu.sg/home/guohua/`