

ON COMPUTABLE SELF-EMBEDDINGS OF COMPUTABLE LINEAR ORDERINGS

RODNEY G. DOWNEY, BART KASTERMANS, AND STEFFEN LEMPP

ABSTRACT. We solve a longstanding question of Rosenstein, and make progress toward solving a long-standing open problem in the area of computable linear orderings by showing that every computable η -like linear ordering without an infinite strongly η -like interval has a computable copy without nontrivial computable self-embedding.

The precise characterization of those computable linear orderings which have computable copies without nontrivial computable self-embedding remains open.

1. INTRODUCTION AND MAIN THEOREM

Computability-theoretic aspects of linear orderings have been the driving force for a number of significant advances in computability theory and computable model theory as well as advances in reverse mathematics. Classical illustrations include Feiner's proof [Fe70] that there are c.e. linear orderings not isomorphic to computable ones (which used codings of a complexity hitherto unseen in applied computability theory), Watnick's [Wa84] codings into discrete linear orderings which formed the basis of jump degrees in linear orderings (Ash-Jockusch-Knight [AJK90]), the Jockusch-Soare [JS91] proof that there are low linear orderings not isomorphic to computable ones (which introduced the notion of a separator and demonstrated that a priority argument could be used to diagonalize against classical isomorphisms), Richter's proof [Ri77] that if an order type has a (least) degree then that degree is $\mathbf{0}$ (which was a canonical example in the theory of degrees of structures), Montalbán's analysis [Mo05, Mo06] of Laver's Theorem which

2000 *Mathematics Subject Classification.* Primary 03D45; Secondary 03C57.

Key words and phrases. computable linear ordering, self-embedding.

The first author's research was supported by The Marsden Fund of New Zealand. The third author's research was supported by NSF grant DMS-0555381 and grant # 13407 by the John Templeton Foundation under its program "Exploring the Infinite". All the authors wish to thank Lempp's doctoral student Dan Turetsky for a careful reading and helpful corrections. The second author thanks VU University Amsterdam for its hospitality during the summer of 2008.

introduced a new set of invariants for linear orderings called signed trees, and the classical example of the Harrison ordering [Ha68] which lies at the heart of basically all proofs of analytic completeness. This list is quite incomplete. We refer the reader to Downey [Do98] and Rosenstein [Ro82] for assorted results in this area.

The concern of this paper is self-embeddings of computable linear orderings. The Dushnik-Miller Theorem [DM41] is a classical result that states that every countable linear ordering $\mathcal{L} = (L, \leq)$ has a nontrivial self-embedding, where a function $f : L \mapsto L$ is called a *self-embedding* of \mathcal{L} iff f is order-preserving and *nontrivial* iff it is not the identity.

The Dushnik-Miller Theorem has given rise to many results in computable model theory. It is easy to see that the result is effectively false in that there is a computable copy of (ω, \leq) with no computable self-embedding (Hay and Rosenstein, see [Ro82, Ro84]). Downey and Moses [DM89] showed that every discrete linear ordering¹ has a computable copy with no strongly nontrivial Π_1^0 -self-embedding, and this was generalized by Downey, Jockusch and Miller [DJM06] to classify the degrees of such self-embeddings in terms of models of Peano Arithmetic. Related here is the theorem of Downey and Lempp [DL99] who proved that the Dushnik-Miller Theorem is equivalent to ACA_0 over RCA_0 , a result which is particularly interesting in that the proof uses a priority argument, but is carried out in the weak system RCA_0 .

In this paper, we will study a long-standing question in the area. It is easy to see that if a computable linear ordering \mathcal{L} has an interval of order type $\sum_{r \in \mathbb{Q}} b_r$ and $b_r \leq n$, that is, \mathcal{L} has an interval which looks like the rationals except that each rational point can be replaced by one or more but at most n points in a block, then each computable copy (isomorphic copy of the order that in its own right is computable) has a nontrivial self-embedding. This embedding can be assembled by fixing the embedding outside of the interval (which without loss of generality has endpoints), and then moving everything within the interval by using the “within n ” density there. The longstanding conjecture of Downey and Moses (see Downey [Do98]) here is that *this is the only way every computable copy has a nontrivial computable self-embedding*.

To be more precise, we have the following definitions.

Definition 1.1. Let $\mathcal{A} = (A, <_{\mathcal{A}})$ be a linear ordering.

- (1) We call two elements $a_0, a_1 \in A$ *finitely far apart* if the interval between a_0 and a_1 is finite (allowing for the case that $a_0 = a_1$). (We denote this equivalence relation by $a_0 \sim^* a_1$.)

¹A *discrete* linear ordering is one where each element has a successor and a predecessor, except for the possible first and last elements.

- (2) The *condensation* of \mathcal{A} is the quotient of \mathcal{A} by \sim^* and denoted by \mathcal{A}^* . We denote the image of $a \in A$ under the quotient map by a^* .

Definition 1.2. Let $\mathcal{A} = (A, <_{\mathcal{A}})$ be a linear ordering. Then \mathcal{A} is called

- η -like² if for any $a \in A$, there are only finitely many elements in A finitely far apart from a (i.e., if each \sim^* -equivalence class is finite, or, equivalently, there are no intervals in \mathcal{A} of order-type ω or ω^*).³
- *strongly η -like* if there is a fixed bound $N \in \omega$ such that for any $a \in A$, there are at most N many elements in A finitely far apart from a (or, equivalently, there are no intervals in \mathcal{A} of order-type $N + 1$).

With this, the conjecture of Downey and Moses becomes:

Conjecture 1.3. *Let \mathcal{L} be an infinite computable linear ordering which does not have an infinite strongly η -like interval. Then there is a computable linear ordering \mathcal{B} isomorphic to \mathcal{A} which has no computable nontrivial self-embedding.*

A few scattered results are known. Showing the depth of the problem, Downey [Do98] showed that there is a computable linear ordering \mathcal{L} such that for any computable linear ordering $\hat{\mathcal{L}}$ isomorphic to \mathcal{L} via a Δ_2^0 -isomorphism, $\hat{\mathcal{L}}$ has a nontrivial computable self-embedding; additionally in [Do98] that there is no uniform method of constructing a computable $\hat{\mathcal{L}}$ isomorphic to \mathcal{L} with no computable self embedding. The reason that these results are of interest is that *most* constructions in computable model theory are uniform and most can be viewed as a kind of game. We will be given some structure \mathcal{A} and construct a structure $\hat{\mathcal{A}}$ with $\mathcal{A} \cong \hat{\mathcal{A}}$ via a stage by stage approximation $f_s : \mathcal{A}_s \mapsto \hat{\mathcal{A}}_s$, where these are stage s approximations to \mathcal{A} and $\hat{\mathcal{A}}$, respectively. Typically, f_s is an isomorphism, and usually it is Δ_2^0 in the sense that $\lim_s f_s(a)$ exists for each $a \in \text{dom } \mathcal{A}$. The game is that the opponent plays some points into the domain (say) and we respond by playing points reflecting the stage s diagram into our version of the range. The hoped-for isomorphism is given by keeping it an isomorphism at every

²Here, η denotes the order type of the rationals.

³In this paper, we will call an *interval* of a linear ordering \mathcal{A} any *convex* subset of \mathcal{A} , i.e., any subset $S \subseteq A$ such that whenever $a, a' \in S$ then any element between a and a' is also in S . So, in particular, under our definition, an interval need not have endpoints in \mathcal{A} .

stage s . Even in cases where f is not Δ_2^0 , there is usually a stage by stage isomorphism, such as in Downey [Do93], where the final mapping is only Δ_3^0 .

In the present paper, we will verify the conjecture for the class of η -like linear orderings. This answers a 24-year-old question of Rosenstein [Ro84, p. 474].

Main Theorem. *Let \mathcal{A} be an infinite η -like computable linear ordering which does not have an infinite strongly η -like interval. Then there is a computable linear ordering \mathcal{B} isomorphic to \mathcal{A} which has no computable nontrivial self-embedding.*

While this might seem rather modest progress towards the full question, η -like orderings were seen as a very important test case. Moreover, the proof of our theorem has, we feel, significant technical interest, which we believe will have wide applications elsewhere. The final isomorphism is *not* Δ_2^0 . *Additionally*, at *no* stage do we ever have a partial isomorphism from the domain to the range.

The remainder of this paper is devoted to the proof of our Main Theorem.

2. INTUITION FOR THE PROOF OF THE MAIN THEOREM

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*.

The key idea of our proof is similar to the proof technique in Downey, Lempp, G. Wu [DLWta], who prove that for any computable linear ordering with infinitely many successivities, there is a computable isomorphic copy in which the successivity relation has degree $\mathbf{0}'$. The idea there was not to try to produce a Δ_2^0 -isomorphism by effectively approximating it by finite partial isomorphisms, but to define finite parts of a Δ_3^0 -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 tries to map one more maximal block of elements in \mathcal{A} to a maximal block in \mathcal{B} of corresponding size.

So let's fix an infinite η -like computable linear ordering $\mathcal{A} = (A, <_{\mathcal{A}})$ which does not have an infinite strongly η -like interval. We need to build a computable linear ordering $\mathcal{B} = (B, <_{\mathcal{B}})$ isomorphic to \mathcal{A} and a (non-computable) map $\iota : A \rightarrow B$, meeting, in increasing order of difficulty, the following four

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{E} : for any computable function $f : B \rightarrow B$ which is not the identity, f is not a self-embedding of \mathcal{B} .

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). (It is probably possible to remove this non-uniformity, but our assumption of no endpoints makes our construction easier.)

The isomorphism ι will now be defined along the true path TP , in the sense that ι is the increasing union of finite partial isomorphisms ι_σ between \mathcal{A} and \mathcal{B} for $\sigma \subset TP$. In the absence of requirements against computable self-embeddings of \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 don't make the "silly" mistake to let ι map elements in \mathcal{A} to elements in \mathcal{B} ordered differently.

2.2. Making ι well-defined. Our technique for meeting this requirement foreshadows the technique we introduce in the next section. We ensure that ι is well-defined not by ensuring it one element at a time, but one maximal block at a time. So fix an element $a \in \mathcal{A}$ for which $\iota(a)$ has not yet been defined (say, such an element with least Gödel number). 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_2^0 -fashion in the sense that there is a computable function $BS : A \times \omega \rightarrow \omega$ such that the size of the maximal block containing a is given by $\liminf_s BS(a, s)$.

Now, at stage s , we simply ensure that there are $BS(a, s)$ many elements in B to which the maximal block of a can be mapped. Whenever $BS(a, s)$ increases (and so in particular when we first start working on the element a), we add more elements to B in order to have a maximal block in B of sufficient size. Whenever $BS(a, s)$ decreases, we discard some of the elements in B previously in the ι -image of the maximal block of a from the current range of ι ; of course, these elements can't be removed from B , but they are now no longer in the range of ι , and the strategies making ι surjective will worry about supplying preimages for these elements of B later on.

2.3. Identifying maximal blocks in \mathcal{A} . We interrupt our discussion to introduce the key idea in how we satisfy the remaining requirements.

The main problem consists in defining the map ι while trying to keep fixed (finite) blocks in \mathcal{B} intact (i.e., not inserting additional elements into them). So suppose we want to keep a block \bar{b} in \mathcal{B} (of length n , say) intact. For this, we need to identify a maximal block \bar{a}' in \mathcal{A} of length $n' \geq n$ (inside some infinite interval of \mathcal{A}) and then let ι map \bar{a}' to a block \bar{b}' containing \bar{b} . We will “guess” the block \bar{a}' in \mathcal{A} as follows: First we guess at a block \bar{a} in \mathcal{A} of length n , looking for one with least Gödel number, discarding all n -tuples of elements in A if they are not $<_{\mathcal{A}}$ -ordered correctly or if there is an extra element in \mathcal{A} in between. For each such \bar{a} , we now guess the length $n' \geq n$ of the maximal block in \mathcal{A} containing \bar{a} , associating with the guess n' each time an n' -tuple \bar{a}' of elements of A which we currently believe to form a maximal block in \mathcal{A} containing \bar{a} .

The *outcomes* of a strategy trying to find a preimage (more generally, inside some infinite interval of \mathcal{A} , say) for a block \bar{b} in \mathcal{B} of length n are thus of the form (\bar{a}, n', \bar{a}') , where \bar{a} is a $<_{\mathcal{A}}$ -ordered n -tuple from (the infinite interval of) \mathcal{A} , $n' \geq n$, and \bar{a}' is a $<_{\mathcal{A}}$ -ordered n' -tuple from \mathcal{A} containing \bar{a} as a block. These outcomes are ordered lexicographically, where the ordering on \bar{a} and \bar{a}' is by Gödel number.

Since no infinite interval of \mathcal{A} is strongly η -like, there are arbitrarily large finite blocks inside each infinite interval of \mathcal{A} . Therefore, each such strategy, looking for a preimage for \bar{b} of length n inside some infinite interval of \mathcal{A} , is guaranteed the existence of a block of length $\geq n$ inside that interval of \mathcal{A} . Furthermore, we can organize the guessing for a “correct” triple (\bar{a}, n', \bar{a}') (i.e., a triple such that \bar{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 (\bar{a}, n', \bar{a}') and triples of the form $(\bar{a}, n'', \bar{a}'')$ (where $n'' > n'$). (This is because if \bar{a} is indeed a block in \mathcal{A} , then there is a block \bar{a}' containing \bar{a} in \mathcal{A} , and once we have correctly guessed this \bar{a}' for the first time (where \bar{a}' may equal \bar{a} if the latter is already a maximal block), we will from then on only be wrong by guessing strictly longer tuples \bar{a}'' to be the maximal block containing \bar{a} .)

The above basic strategy can easily be modified to ensure the satisfaction of our remaining requirements, as we will now explain.

2.4. Making ι surjective. Such a strategy will in general work within an interval of \mathcal{B} , possibly bordered on one or both sides by maximal blocks, and within an infinite interval of \mathcal{A} , again possibly bordered on

one or both sides by maximal blocks. Fix an element b (say, the element with least Gödel number) in some infinite interval of \mathcal{B} which is not currently in the range of ι . We now view b as a 1-tuple of elements of B and apply the strategy from section 2.3 in order to find preimages not only for b , but also for all the elements in a maximal block containing b .

2.5. Defeating one self-embedding. The most complicated type of requirement tries to ensure that no computable function $f : B \rightarrow B$ is a self-embedding of \mathcal{B} . Let's look at a single requirement, i.e., a single potential self-embedding, first, and fix a partial computable function $f : B \rightarrow B$. If f is indeed a nontrivial self-embedding, then there must be an element $b \in B$ such that

- (1) both $f(b)$ and $f(f(b))$ are defined;
- (2) b , $f(b)$ and $f(f(b))$ are pairwise distinct; and
- (3) either $b <_{\mathcal{B}} f(b) <_{\mathcal{B}} f(f(b))$, or $f(f(b)) <_{\mathcal{B}} f(b) <_{\mathcal{B}} b$.

Without loss of generality, we'll assume that

$$b <_{\mathcal{B}} f(b) <_{\mathcal{B}} f(f(b)),$$

since the other case is symmetric.

We now defeat f as a self-embedding of \mathcal{B} by ensuring that there are more elements between b and $f(b)$ than between $f(b)$ and $f(f(b))$, which clearly prevents f from being a self-embedding. When we find b as above, the part of \mathcal{B} defined so far is still finite, of course; so we will

- declare the interval $[f(b), f(f(b))]$ to be fixed from now on, no longer allowing any elements to be inserted in the future; and
- insert sufficiently many new elements (if any) to make sure that already now there are more elements between b and $f(b)$ than between $f(b)$ and $f(f(b))$.

We then apply the technique of section 2.3 to the block $[f(b), f(f(b))]$ (viewed as an n -tuple for some n) in order to find a preimage for each element in $[f(b), f(f(b))]$.

This strategy thus has two types of outcomes: a finitary outcome, denoting that no appropriate element b was found, and infinitary outcomes as outlined in section 2.3, which code the maximal block $[b^0, b^1]$, say, containing $[f(b), f(f(b))]$ and which this strategy is trying to protect in order to defeat f .

There are no constraints on lower-priority strategies assuming the finitary outcome of the strategy defeating f . Lower-priority strategies assuming an infinitary outcome are not allowed to insert any elements into the maximal block $[b^0, b^1]$, but are free to insert elements anywhere else in \mathcal{B} . This, in effect, means that the strategy for f has partitioned \mathcal{B}

into two infinite intervals $(-\infty, b^0)$ and (b^1, ∞) (in addition to the finite interval $[b^0, b^1]$ which no other strategy is allowed to change).

2.6. Defeating two self-embeddings. There are two possibilities for a strategy trying to defeat a function f_1 as a self-embedding of \mathcal{B} below a strategy already having defeated a potential self-embedding f_0 : If the former assumes the finitary outcome of the latter, then the strategy dealing with f_1 can act as if in isolation since there are no higher-priority constraints. In the other case, the strategy dealing with the partial computable function $f_1 : B \rightarrow B$ cannot insert any elements into the interval $[b^0, b^1]$.

But it is not hard to see that if f_1 is indeed a nontrivial self-embedding, then there must be an element $b_1 \in B$ such that

- (1) both $f_1(b_1)$ and $f_1(f_1(b_1))$ are defined;
- (2) b_1 , $f_1(b_1)$ and $f_1(f_1(b_1))$ are pairwise distinct;
- (3) all of b_1 , $f_1(b_1)$ and $f_1(f_1(b_1))$ are either in $(-\infty, b^0)$, or all of b_1 , $f_1(b_1)$ and $f_1(f_1(b_1))$ are in (b^1, ∞) ; and
- (4) either $b_1 <_{\mathcal{B}} f_1(b_1) <_{\mathcal{B}} f_1(f_1(b_1))$, or $f_1(f_1(b_1)) <_{\mathcal{B}} f_1(b_1) <_{\mathcal{B}} b_1$.

We can assume (3) above since if f_1 is indeed a nontrivial self-embedding of \mathcal{B} , moving some element $b \in B$, say, then iterating f_1 on b sufficiently many times will give an element $b_1 = f_1^m(b)$ (for sufficiently large m) satisfying (1)–(4).

Without loss of generality, we'll assume that

$$b_1 <_{\mathcal{B}} f_1(b_1) <_{\mathcal{B}} f_1(f_1(b_1)) <_{\mathcal{B}} b^0,$$

since the other cases are analogous.

The strategy for defeating f_1 is now the obvious one: We will

- declare the interval $[f_1(b_1), f_1(f_1(b_1))]$ to be fixed from now on, no longer allowing any elements to be inserted in the future; and
- insert sufficiently many new elements (if any) to make sure that already now there are more elements between b_1 and $f_1(b_1)$ than between $f_1(b_1)$ and $f_1(f_1(b_1))$.

We then use the technique from section 2.3 to find preimages for a maximal block $[b_1^0, b_1^1]$ containing the interval $[f_1(b_1), f_1(f_1(b_1))]$.

As before, the strategy defeating f_1 has two types of outcomes, a finitary outcome, denoting that no element b_1 as above could be found, as well as infinitely many infinitary outcomes corresponding to various guesses about $[f_1(b_1), f_1(f_1(b_1))]$ and its preimage.

2.7. Defeating several self-embeddings. It should be fairly clear by now how to deal with several higher-priority strategies defeating

various potential self-embeddings: A strategy trying to defeat an additional potential self-embedding given as a partial computable function $f : B \rightarrow B$, say, will be faced with \mathcal{B} partitioned into finitely many infinite intervals I^0, \dots, I^k , say (as well as k many maximal blocks which the strategy is not allowed to change).

For the same reason as in section 2.6, if f is indeed a nontrivial self-embedding, then there must be an element $b \in B$ such that

- (1) both $f(b)$ and $f(f(b))$ are defined;
- (2) $b, f(b)$ and $f(f(b))$ are pairwise distinct;
- (3) all of $b, f(b)$ and $f(f(b))$ are the same interval I^i ; and
- (4) either $b <_{\mathcal{B}} f(b) <_{\mathcal{B}} f(f(b))$, or $f(f(b)) <_{\mathcal{B}} f(b) <_{\mathcal{B}} b$.

Once (if ever) b is found, the strategy will

- declare the interval $[f(b), f(f(b))]$ to be fixed from now on, no longer allowing any elements to be inserted in the future; and
- insert sufficiently many new elements (if any) to make sure that already now there are more elements between b and $f(b)$ than between $f(b)$ and $f(f(b))$.

We then apply the technique of section 2.3 to the block $[f(b), f(f(b))]$ (viewed as an n -tuple for some n) in order to find a preimage for each element in $[f(b), f(f(b))]$.

2.8. Dealing with Strategies to the Left of the True Path.

There is one additional small issue which will arise in the full tree construction and to which we want to alert the reader: When a strategy α , say, wishes to find a preimage for a tuple \bar{b} of elements of B , then α will have to respect not only the finitely many finite intervals given by strategies above α , but also the finitely many finite intervals given by strategies to the left of α . (If α is on the true path, then the strategies to its left will only act finitely often and will thus create only finitely many intervals that α has to deal with.) This is because strategies to the left of α may after all be correct and cannot afford to have their work destroyed during stages when they are “dormant” (i.e., to the left of the current true path).

But since there are only finitely many finite maximal blocks from above and the left that any strategy has to deal with, the strategies described above will still work: We simply have to respect more intervals I^i . Note that this implies also that an \mathcal{S}_b -strategy (for an element b for which $\iota(b)$ is not yet defined) has to find a preimage not only for its element b but also for the entire block of elements in which strategies to the left of the \mathcal{S}_b -strategy believe b to be in (i.e., b may be defined as far as strategies to the left of the \mathcal{S}_b -strategy may be concerned).

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 [So87] or Lempp [LeLN]).

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

$$\Lambda = \{(\bar{a}, n', \bar{a}') \mid 0 < n \leq n', \bar{a} \in A^n, \bar{a}' \in A^{n'}, \bar{a} \subseteq \bar{a}'\} \cup \{fin\}$$

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

We effectively create a list of requirements $\{\mathcal{R}_i\}_{i \in \omega}$ such that each of the following requirements appears in it, for each $a \in A$, $b \in B$, and each partial computable function $f : B \rightarrow B$:

\mathcal{W}_a : $\iota(a)$ is (well-)defined

\mathcal{S}_b : $\iota^{-1}(b)$ is defined

\mathcal{E}_f : $f \neq \text{id}_B \Rightarrow f$ is not a self-embedding of \mathcal{B}

All strategies $\sigma \in T$ of length i are assigned to requirement \mathcal{R}_i .

3.2. The full strategies. Each strategy is equipped with a finite partial map $\iota_\sigma : A \rightarrow B$, which is its current guess about the isomorphism $\iota : \mathcal{A} \rightarrow \mathcal{B}$. 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 strategy is also associated with a “block size” function, approximated by a computable function $BS_\sigma : A \times \omega \rightarrow \omega$, which, for all σ along the true path TP , will have the property that

$$(1) \quad \begin{aligned} BS_\sigma(a) &:= \liminf_{s \in S_\sigma} BS_\sigma(a, s) \\ &= \text{size of the maximal block in } \mathcal{A} \text{ containing } a \end{aligned}$$

For this, we define two auxiliary functions, for two consecutive stages s' and s in S_σ :

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

Intuitively, the functions L_σ and R_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, but they may not drop to the true \liminf at the same stages. We circumvent this problem by defining

$$BS_\sigma(a, s) = L_\sigma(a, s) + 1 + \min \{ R(a, s') \mid s' \leq s \text{ and } s' \in S_\sigma \text{ and} \\ \forall s'' \in (s', s) \cap S_\sigma (L_\sigma(a, s'') > L_\sigma(a, s)) \}.$$

It is now easy to check that this definition of the function BS_σ ensures (1) above for all strategies σ along the true path (and indeed for all strategies σ for which S_σ is infinite).

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 at which σ is not delayed (as defined at the end of section 3.3). (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 $L_\sigma(a, s)$ many elements currently immediately to the left of a as well as the $BS_\sigma(a, s) - L_\sigma(a, s) - 1$ many elements currently immediately to the right of a ; we'll denote this tuple of elements in A by \vec{a}' . The outcome of the strategy is now $(a, BS_\sigma(a, s), \vec{a}')$ (denoting that the strategy guesses that the maximal block containing a consists of the $BS_\sigma(a, s)$ many elements in \vec{a}'). Then the strategy defines $\iota_\sigma(\vec{a}')$ 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 create $BS_\sigma(a, s)$ many

new elements \bar{b}' in B and $<_{\mathcal{B}}$ -order them consistently with ι_{σ}^{-} .
 Declare that $\iota_{\sigma}(\bar{a}') = \bar{b}'$.

- If $s' < s$ and $L(a, s') < L_{\sigma}(a, s)$, then create $L_{\sigma}(a, s) - L_{\sigma}(a, s')$ many new elements in B immediately to the left of $\iota_{\sigma, s'}(\bar{a}'')$, where (a, n'', \bar{a}'') was the outcome of σ at stage s' .
- If $s' < s$ and $BS(a, s') - L_{\sigma}(a, s') < BS(a, s) - L_{\sigma}(a, s)$, then create $(BS(a, s) - L_{\sigma}(a, s)) - (BS(a, s') - L_{\sigma}(a, s'))$ many new elements in B immediately to the right of $\iota_{\sigma, s'}(\bar{a}'')$, where (a, n'', \bar{a}'') was the outcome of σ at stage s' .
- Denote by \bar{a}' the tuple in A consisting of the $L_{\sigma}(a, s)$ many elements now immediately to the left of a , a itself, as well the $BS_{\sigma}(a, s) - L_{\sigma}(a, s) - 1$ many elements now immediately to the right of a , and by \bar{b}' the tuple in B consisting of the $L_{\sigma}(a, s)$ many elements now immediately to the left of b , b itself, as well the $BS_{\sigma}(a, s) - L_{\sigma}(a, s) - 1$ many elements now immediately to the right of b . Declare that $\iota_{\sigma}(\bar{a}') = \bar{b}'$.

The strategy now ends the substage with outcome $(a, BS_{\sigma}(a, s), \bar{a}')$.

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, check first whether b is part of a tuple restrained by an \mathcal{E} -strategy $\tau <_L \sigma$. If not, then set $\bar{b} = b$. Otherwise, let \bar{b} be the shortest tuple of elements in B containing b such that no \mathcal{E} -strategy $\tau <_L \sigma$ restrains both elements in \bar{b} and elements outside \bar{b} . In either case, we let n be the length of \bar{b} .

Now the strategy finds the tuple $\bar{a} \in (A - \text{dom}(\iota_{\sigma}^{-}))^n$ (with least Gödel number) which is $<_{\mathcal{A}}$ -ordered with respect to $\text{dom}(\iota_{\sigma}^{-})$ as \bar{b} is $<_{\mathcal{B}}$ -ordered with respect to $\text{ran}(\iota_{\sigma}^{-})$, and we let a be the element in \bar{a} in the same position as b is in \bar{b} . (If currently no such \bar{a} exists, then the strategy ends the substage with outcome *fin*; this delay must be finite if the strategy is on the true path.) Now the strategy guesses that the maximal block containing \bar{a} also contains the $L_{\sigma}(a, s)$ many elements currently immediately to the left of a as well as the $BS_{\sigma}(a, s) - L_{\sigma}(a, s) - 1$ many elements currently immediately to the right of a ; we'll denote this tuple of elements in A by \bar{a}' . The outcome of the strategy is now $(\bar{a}, BS_{\sigma}(a, s), \bar{a}')$ (denoting that the strategy guesses that the maximal block containing \bar{a} consists of the $BS_{\sigma}(a, s)$ many elements in \bar{a}'). Then the strategy defines $\iota_{\sigma}(\bar{a}')$ 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 create $BS_{\sigma}(a, s) - 1$

many new elements in B around b and denote by \bar{b}' the new elements of B together with b (such that a is the m th element of \bar{a}' iff b is the m th element of \bar{b}' and such that \bar{b}' forms a block in \mathcal{B}). Declare that $\iota_\sigma(\bar{a}') = \bar{b}'$.

- If $s' < s$ and $L(a, s') < L_\sigma(a, s)$, then create $L_\sigma(a, s) - L_\sigma(a, s')$ many new elements in B immediately to the left of $\iota_{\sigma, s'}(\bar{a}'')$, where $(\bar{a}, n'', \bar{a}'')$ was the outcome of σ at stage s' .
- If $s' < s$ and $BS(a, s') - L_\sigma(a, s') < BS(a, s) - L_\sigma(a, s)$, then create $(BS(a, s) - L_\sigma(a, s)) - (BS(a, s') - L_\sigma(a, s'))$ many new elements in B immediately to the right of $\iota_{\sigma, s'}(\bar{a}'')$, where $(\bar{a}, n'', \bar{a}'')$ was the outcome of σ at stage s' .
- Denote by \bar{a}' the tuple in A consisting of the $L_\sigma(a, s)$ many elements now immediately to the left of a , a itself, as well the $BS_\sigma(a, s) - L_\sigma(a, s) - 1$ many elements now immediately to the right of a , and by \bar{b}' the tuple in B consisting of the $L_\sigma(a, s)$ many elements now immediately to the left of b , b itself, as well the $BS_\sigma(a, s) - L_\sigma(a, s) - 1$ many elements now immediately to the right of b . Declare that $\iota_\sigma(\bar{a}') = \bar{b}'$.

The strategy now ends the substage with outcome $(a, BS_\sigma(a, s), \bar{a}')$.

3.2.3. *The full \mathcal{E}_f -strategy.* This strategy σ has to ensure that f is not a nontrivial self-embedding of \mathcal{B} .

If $s = s'$ or if the outcome of the strategy at stage s' was *fin*, then the strategy checks whether there is $b \in B - \text{ran}(\iota_\sigma^-)$ such that

- (1) both $f(b)$ and $f(f(b))$ are defined;
- (2) b , $f(b)$ and $f(f(b))$ are pairwise distinct;
- (3) either $b <_{\mathcal{B}} f(b) <_{\mathcal{B}} f(f(b))$, or $f(f(b)) <_{\mathcal{B}} f(b) <_{\mathcal{B}} b$; and
- (4) there is no element from

$$\text{ran}(\iota_\sigma^-) \cup \bigcup_{\tau < L\sigma} \text{ran}(\iota_\tau)$$

in $[b, f(f(b))]$ or $[f(f(b)), b]$, respectively.

If there is no such b , then the strategy simply ends the substage with outcome *fin*. Otherwise, by symmetry, assume that $b <_{\mathcal{B}} f(b) <_{\mathcal{B}} f(f(b))$. Denote by \bar{b} the tuple of elements in $[f(b), f(f(b))]$, letting n be the size of this interval. Now the strategy

- (1) inserts sufficiently many new elements into \mathcal{B} so that the interval $[b, f(b)]$ has more than n many elements;
- (2) finds an n -tuple of adjacent elements $\bar{a} \in (A - \text{dom}(\iota_\sigma^-))^n$ (with least Gödel number) which is $<_{\mathcal{A}}$ -ordered with respect to $\text{dom}(\iota_\sigma^-)$ as \bar{b} is $<_{\mathcal{B}}$ -ordered with respect to $\text{ran}(\iota_\sigma^-)$;

- (3) guesses that the maximal block containing a_1 (the $<_{\mathcal{A}}$ -least element of \bar{a}) also contains the $L_{\sigma}(a_1, s)$ many elements currently immediately to the left of a_1 as well as the

$$\min\{n - 1, BS_{\sigma}(a_1, s) - L_{\sigma}(a_1, s) - 1\}$$

many elements currently immediately to the right of a_1 (we'll denote this tuple of elements in A by \bar{a}' , and let n' be its length);

- (4) inserts $n' - n$ many new elements into \mathcal{B} just to the right of $f(f(b))$ (we'll denote by \bar{b}' the elements of $[f(b), f(f(b))]$ together with these new elements);
- (5) declares that $\iota_{\sigma}(\bar{a}') = \bar{b}'$; and
- (6) ends the substage with outcome (\bar{a}, n', \bar{a}') (denoting that the strategy guesses that the maximal block containing \bar{a} consists of the $BS_{\sigma}(a_1, s)$ many elements in \bar{a}').

Otherwise, i.e., if $s' < s$ and such b had already been found by stage s' , let $(\bar{a}'', n''', \bar{a}''')$ be the strategy's outcome at stage s' . The strategy then

- (1) finds an n -tuple of adjacent elements $\bar{a} \in (A - \text{dom}(\iota_{\sigma}^-))^n$ (with least Gödel number) which is $<_{\mathcal{A}}$ -ordered with respect to $\text{dom}(\iota_{\sigma}^-)$ as \bar{b} is $<_{\mathcal{B}}$ -ordered with respect to $\text{ran}(\iota_{\sigma}^-)$;
- (2) guesses that the maximal block containing a_1 (the $<_{\mathcal{A}}$ -least element of \bar{a}) also contains the $L_{\sigma}(a_1, s)$ many elements currently immediately to the left of a_1 as well as the

$$\min\{n - 1, BS_{\sigma}(a_1, s) - L_{\sigma}(a_1, s) - 1\}$$

many elements currently immediately to the right of a_1 (we'll denote this tuple of elements in A by \bar{a}' , and let n' be its length);

- (3) if $s' < s$ and $L(a_1, s') < L_{\sigma}(a_1, s)$, creates $L_{\sigma}(a_1, s) - L_{\sigma}(a_1, s')$ many new elements in B immediately to the left of $\iota_{\sigma, s'}(\bar{a}'')$, where $(\bar{a}, n'', \bar{a}'')$ was the outcome of σ at stage s' ;
- (4) if $s' < s$ and $BS(a_1, s') - L_{\sigma}(a_1, s') < BS(a_1, s) - L_{\sigma}(a_1, s)$, creates $(BS(a_1, s) - L_{\sigma}(a_1, s)) - (BS(a_1, s') - L_{\sigma}(a_1, s'))$ many new elements in B immediately to the right of $\iota_{\sigma, s'}(\bar{a}'')$, where $(\bar{a}, n'', \bar{a}'')$ was the outcome of σ at stage s' ;
- (5) denotes by \bar{a}' the tuple in A consisting of the $L_{\sigma}(a_1, s)$ many elements now immediately to the left of a_1 , a_1 itself, as well as the $BS_{\sigma}(a_1, s) - L_{\sigma}(a_1, s) - 1$ many elements now immediately to the right of a_1 , and by \bar{b}' the tuple in B consisting of the $L_{\sigma}(a_1, s)$ many elements now immediately to the left of $f(b)$, $f(b)$ itself, as well as the $BS_{\sigma}(a_1, s) - L_{\sigma}(a_1, s) - 1$ many elements now immediately to the right of $f(b)$;

- (6) declares that $\iota_\sigma(\bar{a}') = \bar{b}'$; and
- (7) ends the substage with outcome (\bar{a}, n', \bar{a}') (denoting that the strategy guesses that the maximal block containing \bar{a} consists of the $BS_\sigma(a_1, s)$ many elements in \bar{a}').

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 $t < s$, the strategy σ will determine its outcome o and thus the strategy $\sigma \hat{\ } \langle o \rangle$ eligible to act at the next substage. The action of each strategy is determined by the type of requirement it is assigned to and is as described in section 3.2.

If a strategy σ tries to define a block \bar{a}' which intersects with $\text{dom}(\iota_\sigma^-)$, then we call the strategy *delayed*, and the strategy will take no action and instead end the substage with outcome *fin*. (Such a delay must be finite if the strategy is along the true path.)

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 verify the satisfaction of all requirements in a sequence of lemmas, starting with the usual properties of a priority argument on a tree of strategies.

Lemma 3.1. *There is a leftmost path through T of strategies (called the true path), i.e., we can inductively define an infinite path $TP \in [T]$ such that for all m :*

- (1) $TP \upharpoonright m$ is eligible to act infinitely often; and
- (2) any $\sigma <_L TP \upharpoonright m$ is eligible to act at most finitely often.

Proof. We proceed by induction on m . The case $m = 0$ is trivial; so suppose that $\sigma = TP \upharpoonright m$ has been defined and satisfies (1) and (2).

First of all, note that σ cannot be delayed at cofinitely many of the stages at which it is eligible to act. To see this, note that at any stage,

- the domain of ι_σ^- consists of finitely many blocks,
- the finite number k , say, of these blocks eventually stabilizes, and
- each of these blocks eventually stabilizes to a maximal block of \mathcal{A} .

Since \mathcal{A} is η -like without endpoints, the left and right endpoints of each such maximal block are left and right limit-points, respectively; so each delay is finite. Furthermore, these maximal blocks break up \mathcal{A} into finitely many infinite intervals I^0, \dots, I^k between these maximal blocks (as well as all the way to the left and right).

If the outcome of σ is *fin* at cofinitely many stages at which σ is eligible to act, then $TP \upharpoonright (m+1) = \sigma \hat{\ } \langle \text{fin} \rangle$ trivially satisfies (1) and (2).

Otherwise, starting at the least stage s_0 , say, after which σ is no longer initialized, σ will work with a fixed 1-tuple a (if σ is a \mathcal{W} -strategy), or search for a tuple \bar{a} in a fixed interval I^i , say (if σ is an \mathcal{S} -strategy), or search for a tuple \bar{a} in any interval I^i (if σ is an \mathcal{E} -strategy). By the fact that \mathcal{A} is not strongly η -like, σ will eventually work with a fixed tuple \bar{a} of length n , say (which has least Gödel number among the tuples satisfying its search criteria if σ is an \mathcal{E} - or \mathcal{S} -strategy). This tuple is contained inside \mathcal{A} in a maximal block \bar{a}' of length $n' \geq n$, say. Fix a stage $s_1 \geq s_0$ at which all elements of \bar{a}' have appeared; from then on, σ will only have outcome $(\bar{a}, n'', \bar{a}'')$ where $n'' \geq n'$ and $\bar{a}'' \supseteq \bar{a}'$, and whenever $n'' = n'$ then $\bar{a}'' = \bar{a}'$. Parts (1) and (2) of the lemma now follow immediately. \square

We note an important fact about outcomes from the paragraph above in a separate lemma:

Lemma 3.2. *If $\tau = \sigma \hat{\ } \langle o \rangle \subset TP$ and $o = (\bar{a}, n', \bar{a}')$ then at almost all stages $s \in S_\sigma$, σ has outcome (\bar{a}, n', \bar{a}') or an outcome $(\bar{a}, n'', \bar{a}'')$ where $n'' > n'$ and $\bar{a}'' \supset \bar{a}'$. \square*

We now state some preliminary properties of the map ι as constructed along the true path.

Lemma 3.3. *Let $\tau = \sigma \hat{\ } \langle o \rangle \subset TP$. Then*

- (1) *at all stages at which τ is eligible to act, $\iota_\sigma = \iota_\tau^- \subseteq \iota_\tau$; and*
- (2) *$\lim_{s \in S_\tau} \iota_\sigma$ exists and is a finite partial isomorphism from \mathcal{A} into \mathcal{B} .*

Proof. Part (1) is immediate by the definition of ι_τ^- and ι_τ . Also, at any σ -stage, ι_σ is clearly a finite partial isomorphism (i.e., an order-preserving map and thus 1–1) from \mathcal{A} into \mathcal{B} .

By definition, at all stages $s \in S_\tau$, σ has its true outcome (namely, the outcome o with $\sigma \hat{\ } \langle o \rangle = \tau$). So for part (2), it only remains to be shown that $\lim_{s \in S_\tau} \iota_\sigma$ exists. For this, first note that the domain of ι_σ is completely determined by τ and its guesses about tuples in \mathcal{A} .

In order to show that ι_σ stabilizes at τ -stages, note that, by induction, we are done in case $o = \text{fin}$ (since then $\iota_\sigma^- = \iota_\sigma$ at τ -stages), and

if $o = (\bar{a}, n', \bar{a}')$, we only need to show that $\iota_\sigma(\bar{a}')$ reaches a limit over τ -stages. But this is immediate from Lemma 3.2 and the description of the full strategies in section 3.2. \square

Lemma 3.3 implies in particular that ι is a well-defined (possibly partial) order isomorphism from \mathcal{A} into \mathcal{B} . The next lemma shows that ι is indeed an isomorphism.

Lemma 3.4. *The map*

$$\iota = \bigcup_{\sigma \subset TP} \iota_\sigma$$

is an order isomorphism from \mathcal{A} onto \mathcal{B} .

Proof. To show totality of ι , fix $a \in A$. Fix a \mathcal{W}_a -strategy $\sigma \subset TP$ and the tuple $\bar{a}' \subset A$ constituting the maximal block (in order, of length n' , say) containing a . Then σ has true outcome (a, n', \bar{a}') , and since σ cannot be delayed indefinitely, it will eventually find $\bar{b}' \in B^{n'}$ and will map \bar{a}' to \bar{b}' at all but finitely many τ -stages (where $\tau = \sigma \hat{\ } \langle (a, n', \bar{a}') \rangle$).

To show that ι is surjective, fix $b \in B$. Fix an \mathcal{S}_b -strategy $\sigma \subset TP$ (or a strategy $\sigma \subset TP$ above it already determining the preimage of b as part of its own strategy). Then σ will first choose a preimage $a \in A$, then the maximal block \bar{a}' in \mathcal{A} of length n' , say, containing a , and then an image \bar{b}' for \bar{a}' . Then σ has true outcome (a, n', \bar{a}') (for this $a \in A$ and $\bar{a}' \in A^{n'}$) and will map \bar{a}' to \bar{b}' at all but finitely many τ -stages (where again $\tau = \sigma \hat{\ } \langle (a, n', \bar{a}') \rangle$). \square

It is easy to see from our construction that \mathcal{B} is indeed a computable linear ordering, which by Lemma 3.4 is isomorphic to \mathcal{A} . So all we're left to show is that \mathcal{B} does not have any nontrivial computable self-embedding:

Lemma 3.5. *Let $f : B \rightarrow B$ be a computable map. Then f is not a nontrivial self-embedding.*

Proof. Fix any (total) computable map $F : B \rightarrow B$. Fix an \mathcal{E}_f -strategy $\sigma \subset TP$. As in the proof of Lemma 3.1, we can fix finitely many infinite intervals I^0, \dots, I^k in \mathcal{A} such that the domain of ι_σ^- consists exactly of the maximal blocks \bar{a}_i (for $1 \leq i \leq k$) between these intervals. By Lemma 3.3, for each $i \in [1, k]$, we can fix \bar{b}_i such that at all but finitely many σ -stages, $\iota_\sigma^-(\bar{a}_i) = \bar{b}_i$. Furthermore, there will be finitely many tuples \bar{b}^i which σ has to respect for the sake of strategies to its left.

Now suppose that f is a nontrivial self-embedding of \mathcal{B} . Then we can fix $b \in B$ (with least Gödel number) such that $b, f(b)$ and $f(f(b))$ are pairwise distinct, are all in the same interval I^i , and such that there

is no \bar{b}^i between them. By symmetry, we'll assume that $b <_{\mathcal{B}} f(b) <_{\mathcal{B}} f(f(b))$.

But if f were a nontrivial self-embedding, then this would contradict the construction as described in section 3.2.3: If such b could be found, then σ would freeze the interval $[f(b), f(f(b))]$ (making it finite of size n , say) and insert sufficiently many elements into the interval $[b, f(b)]$ so that it contains more than n many elements. \square

REFERENCES

- [AJK90] Ash, Christopher J.; Jockusch, Carl G., Jr.; and Knight, Julia F., *Jumps of orderings*, Trans. Amer. Math. Soc. **319** (1990), 573–599.
- [Do93] Downey, Rodney G., *Every recursive Boolean algebra is isomorphic to one with incomplete atoms*, Ann. Pure Appl. Logic **60** (1993), 193–206.
- [Do98] Downey, Rodney G., *Computability theory and linear orderings*, Handbook of recursive mathematics, Vol. 2, 823–976, North-Holland, Amsterdam, 1998.
- [DJM06] Downey, Rodney G.; Jockusch, Carl G., Jr.; and Miller, Joseph S., *On self-embeddings of computable linear orderings*, Ann. Pure Appl. Logic **138** (2006), 52–76.
- [DL99] Downey, Rodney G.; and Lempp, Steffen, *The proof-theoretic strength of the Dushnik-Miller Theorem for countable linear orders*, “Recursion theory and complexity” (Kazan, 1997), 55–57, de Gruyter, Berlin, 1999.
- [DLWta] Downey, Rodney G.; Lempp, Steffen and Wu, Guohua, *On the complexity of the successivity relation in computable linear orderings*, to appear.
- [DM89] Downey, Rodney G. and Moses, Michael F., *On choice sets and strongly nontrivial self-embeddings of recursive linear orders*, Z. Math. Logik Grundlag. Math. **35** (1989), 237–246.
- [DM41] Dushnik, Ben and Miller, E. W., *Partially ordered sets*, Amer. J. of Math. **63** (1941), 600610.
- [Fe70] Feiner, Lawrence, *Hierarchies of Boolean algebras*, J. Symbolic Logic **35** (1970), 365–374.
- [Ha68] Harrison, Joseph, *Recursive pseudo-well-orderings*, Trans. Amer. Math. Soc. **131** (1968), 526–543.
- [JS91] Jockusch, Carl G., Jr.; and Soare, Robert I., *Degrees of orderings not isomorphic to recursive linear orderings*, “International Symposium on Mathematical Logic and its Applications” (Nagoya, 1988), Ann. Pure Appl. Logic **52** (1991), 39–64.
- [LeLN] Lempp, Steffen, *Lecture Notes on Priority Arguments*, preprint available at <http://www.math.wisc.edu/~lempp/papers/prio.pdf>.
- [Mo05] Montalbán, Antonio, *Beyond the Arithmetic*, Ph.D. Thesis, Cornell University, 2005.
- [Mo06] Montalbán, Antonio, *Countably complementable linear orderings*, Order **23** (2006), 321–331.
- [Ri77] Richter, Linda Jean C., *Degrees of unsolvability of models*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1977.
- [Ro82] Rosenstein, Joseph G., *Linear orderings*, Academic Press, New York-London, 1982.

- [Ro84] Rosenstein, Joseph G., *Recursive linear orderings*, “Orders: description and roles” (L’Arbresle, 1982), 465–475, North-Holland, Amsterdam, 1984.
- [So87] Soare, Robert I., *Recursively enumerable sets and degrees*, Springer-Verlag, Berlin, New York, 1987.
- [Wa84] Watnick, Richard, *A generalization of Tennenbaum’s theorem on effectively finite recursive linear orderings*, *J. Symbolic Logic* **49** (1984), 563–569.

SCHOOL OF MATHEMATICS, STATISTICS AND COMPUTER SCIENCE, VICTORIA
UNIVERSITY, WELLINGTON 6140, NEW ZEALAND

E-mail address: Rod.Downey@mcs.vuw.ac.nz

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF COLORADO, MADISON, WI
80309-0395

E-mail address: bart.kastermans@colorado.edu

E-mail address: lempp@math.wisc.edu