

# ON COMPUTABLE SELF-EMBEDDINGS OF COMPUTABLE LINEAR ORDERINGS

RODNEY G. DOWNEY, BART KASTERMANS, AND STEFFEN LEMPP

ABSTRACT. We make progress toward solving a long-standing open problem in the area of computable linear orderings by showing that every computable  $\eta$ -like linear ordering without an infinite strongly  $\eta$ -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

Introduction...

**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  $a_0 \sim^* a_1$ .)
- (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

- *$\eta$ -like* 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  $\omega$  or  $\omega^*$ ).<sup>1</sup>

---

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 a grant by the Templeton Foundation under its program "Exploring the Infinite".

<sup>1</sup>In 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  $\mathcal{A}$ .

- *strongly  $\eta$ -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  $N + 1$ ).

In an  $\eta$ -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*.

**Definition 1.3.** Let  $\mathcal{A} = (A, <_{\mathcal{A}})$  be a linear ordering. A map  $f : A \rightarrow A$  is called

- a *self-embedding* if  $f$  is order-preserving, i.e., if for all  $a, a' \in A$ ,  $a <_{\mathcal{A}} a'$  implies  $f(a) <_{\mathcal{A}} f(a')$  (and so in particular  $f$  is injective);
- a *nontrivial* self-embedding if in addition  $f(a) \neq a$  for some  $a \in A$ ; and
- a *strongly nontrivial* self-embedding if for some  $a \in A$ , there are infinitely many elements in  $A$  between  $a$  and  $f(a)$ .

A long-standing conjecture in the area of computable linear orderings can now be stated as follows:

**Conjecture 1.4.** *Let  $\mathcal{A}$  be an infinite computable linear ordering which does not have an infinite strongly  $\eta$ -like interval. Then there is a computable linear ordering  $\mathcal{B}$  isomorphic to  $\mathcal{A}$  which has no computable nontrivial self-embedding.*

The main result of this paper is a partial solution to this conjecture:

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

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

## 2. INTUITION FOR THE PROOF OF THE MAIN THEOREM

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  $\Delta_2^0$ -isomorphism by effectively approximating it by finite partial isomorphisms, but to define finite parts of a  $\Delta_3^0$ -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  $\eta$ -like computable linear ordering  $\mathcal{A} = (A, <_{\mathcal{A}})$  which does not have an infinite strongly  $\eta$ -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:*

- $\mathcal{O}$ :  $\iota$  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  $\iota$  is injective);
- $\mathcal{W}$ :  $\iota$  is well-defined (and in particular total);
- $\mathcal{S}$ :  $\iota$  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  $\eta$ -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 elements from an  $\eta$ -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  $\iota$  will now be defined along the true path  $TP$ , in the sense that  $\iota$  is the increasing union of finite partial isomorphisms  $\iota_{\sigma}$  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  $\iota$ , with an eye toward later extending it to the full construction.

**2.1. Making  $\iota$  order-preserving.** This is the simplest requirement: We simply ensure that we don't make the "silly" mistake to let  $\iota$  map elements of  $A$  to elements in  $B$  ordered differently.

**2.2. Making  $\iota$  well-defined.** Our technique for meeting this requirement foreshadows the technique we introduce in the next section. We ensure that  $\iota$  is well-defined not by ensuring it one element at a time, but one maximal block at a time. So fix an element  $a \in A$  for which  $\iota(a)$  has not yet been defined (say, the 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  $\Pi_2^0$ -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  $\iota$ -image of the maximal block of  $a$  from the current range of  $\iota$ ; of course, these elements can't be removed from  $B$ , but they are now no longer in the range of  $\iota$ , and the strategies making  $\iota$  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  $\iota$  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  $\iota$  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  $\eta$ -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  $(\bar{a}, n', \bar{a}')$ . (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  $\iota$  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 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  $\iota$ . 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  $(b^1, \infty)$  (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  $(b^1, \infty)$ ; 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  $b_1 = f^m(b)$  (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  $[f(b), f(f(b))]$ .

### 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 [?]).

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

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

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  $\sigma$  will always have to live with

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

Furthermore, we define, for each strategy  $\sigma$ , the *set  $S_\sigma$  of stages at which  $\sigma$  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 3.2).

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  $\sigma$  along the true path  $TP$ , will have the property that

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

For this, we define two auxiliary functions, for two consecutive stages  $s'$  and  $s$  in  $S_\sigma$ :

$L_\sigma(a, s) = |[a', a]| - 1$ , where  $a'$  is  $<_{\mathcal{A}}$ -least such that no element

has entered  $\mathcal{A}$  in the interval  $[a', a]$  since stage  $s'$ , and

$R_\sigma(a, s) = |[a, a'']| - 1$ , 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  $L_\sigma$  and  $R_\sigma$  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  $BS_\sigma$  ensures (??) above for all strategies  $\sigma$  along the true path (and indeed for all strategies  $\sigma$  for which  $S_\sigma$  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  $S_\sigma$  since  $\sigma$ 's most recent initialization at which  $\sigma$  is not delayed (as defined at the end of section 3.2). (We set  $s' = s$  if no such stage exists.)

3.2.1. *The full  $\mathcal{W}_a$ -strategy.* This strategy 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  $\sigma$  is eligible to act since its most recent initialization), then create  $BS(a, s)$  many new elements  $\vec{b}'$  in  $B$  and  $<_B$ -order them consistently with  $\iota_\sigma^-$ . Declare that  $\iota_\sigma(\vec{a}') = \vec{b}'$ .
- If  $s' < s$  and  $BS(a, s') < BS(a, s)$ , then create  $BS(a, s) - BS(a, s')$  many new elements in  $B$  immediately to the right of  $\iota_{\sigma, s'}(\vec{a}'')$ , where  $(a, n'', \vec{a}'')$  was the outcome of  $\sigma$  at stage  $s'$ . Denote by  $\vec{a}'$  the tuple in  $A$  consisting of the  $L_\sigma(a, s)$  many elements currently immediately to the left of  $a$ ,  $a$  itself, as well as the  $BS_\sigma(a, s) - L_\sigma(a, s) - 1$  many elements currently immediately

to the right of  $a$ , and by  $\vec{b}'$  the elements in the tuple  $\iota_{\sigma,s'}(\vec{a}'')$  together with the new elements of  $B$ . Declare that  $\iota_{\sigma}(\vec{a}') = \vec{b}'$ .

- Otherwise, denote by  $\vec{a}'$  the tuple in  $A$  consisting of the  $L_{\sigma}(a, s)$  many elements currently immediately to the left of  $a$ ,  $a$  itself, as well as the  $BS_{\sigma}(a, s) - L_{\sigma}(a, s) - 1$  many elements currently immediately to the right of  $a$ , and denote by  $\vec{b}'$  the elements in the tuple  $\iota_{\sigma,s'}(\vec{a}')$ . Declare that  $\iota_{\sigma}(\vec{a}') = \vec{b}'$ .

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

**3.2.2. The full  $\mathcal{S}_b$ -strategy.** This strategy 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 finds the element  $a \in A - \text{dom}(\iota_{\sigma}^{-})$  (with least Gödel number) which is  $<_{\mathcal{A}}$ -ordered with respect to  $\text{dom}(\iota_{\sigma}^{-})$  as  $b$  is  $<_{\mathcal{B}}$ -ordered with respect to  $\text{ran}(\iota_{\sigma}^{-})$ . (If currently no such  $a$  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  $a$  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  $\sigma$  is eligible to act since its most recent initialization), then create  $BS(a, s) - 1$  many new elements in  $B$  around  $b$  and denote by  $\vec{b}'$  the new elements of  $B$  together with  $b$  (such that  $a$  is the  $m$ th element of  $\vec{a}'$  iff  $b$  is the  $m$ th element of  $\vec{b}'$  and such that  $\vec{b}'$  forms a block in  $\mathcal{B}$ ). Declare that  $\iota_{\sigma}(\vec{a}') = \vec{b}'$ .
- If  $s' < s$  and  $BS(a, s') < BS(a, s)$ , then create  $BS(a, s) - BS(a, s')$  many new elements in  $B$  immediately to the right of  $\iota_{\sigma,s'}(\vec{a}'')$  where  $(a, n'', \vec{a}'')$  was the outcome of  $\sigma$  at stage  $s'$ . Denote by  $\vec{a}'$  the tuple in  $A$  consisting of the  $L_{\sigma}(a, s)$  many elements currently immediately to the left of  $a$ ,  $a$  itself, as well as the  $BS_{\sigma}(a, s) - L_{\sigma}(a, s) - 1$  many elements currently immediately to the right of  $a$ , and by  $\vec{b}'$  the elements in the tuple  $\iota_{\sigma,s'}(\vec{a}'')$  together with the new elements of  $B$ . Declare that  $\iota_{\sigma}(\vec{a}') = \vec{b}'$ .
- Otherwise, denote by  $\vec{a}'$  the tuple in  $A$  consisting of the  $L_{\sigma}(a, s)$  many elements currently immediately to the left of  $a$ ,  $a$  itself,

as well as the  $BS_\sigma(a, s) - L_\sigma(a, s) - 1$  many elements currently immediately to the right of  $a$ , and by  $\bar{b}'$  the elements in the tuple  $\iota_{\sigma, s'}(\bar{a}')$ . 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 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  $\text{ran}(\iota_\sigma^-)$  between  $b, f(b)$ , and  $f(f(b))$ .

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 the 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  $BS_\sigma(a_1, s) > BS_\sigma(a_1, s')$ , inserts  $BS_\sigma(a_1, s) - BS_\sigma(a_1, s')$  many new elements into  $\mathcal{B}$  just to the right of  $\iota_{\sigma, s'}^-(\bar{a}''')$  (and we'll denote by  $\bar{b}'$  the elements of  $\iota_{\sigma, s'}^-(\bar{a}''')$  together with the new elements);
- (4) if  $BS_\sigma(a_1, s) \leq BS_\sigma(a_1, s')$ , lets  $\bar{b}' = \iota_{\sigma, s'}^-(\bar{a}')$ ;
- (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  $\bar{a}'$ ).

**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  $\sigma$  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  $\sigma$  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 3.1.

If a strategy  $\sigma$  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 (??) and (??).

First of all, note that  $\sigma$  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  $\iota_\sigma^-$  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  $\eta$ -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 block 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  $\sigma$  is *fin* at cofinitely many stages at which  $\sigma$  is eligible to act, then  $TP \upharpoonright (m+1) = \sigma \hat{\ } \langle \text{fin} \rangle$  trivially satisfies (??) and (??).

Otherwise, starting at the least stage  $s_0$ , say, after which  $\sigma$  is no longer initialized,  $\sigma$  will work with a fixed 1-tuple  $\bar{a}$  (if  $\sigma$  is a  $\mathcal{W}$ -strategy), or search for a 1-tuple  $\bar{a}$  in a fixed interval  $I^i$ , say (if  $\sigma$  is an  $\mathcal{S}$ -strategy), or search for a tuple  $\bar{a}$  in any interval  $I^i$  (if  $\sigma$  is an  $\mathcal{E}$ -strategy). By the fact that  $\mathcal{A}$  is not strongly  $\eta$ -like,  $\sigma$  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  $\sigma$  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,  $\sigma$  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 (??) and (??) 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  $s \in S_\sigma$ ,  $\sigma$  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  $\iota$  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  $\tau$  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 (??) is immediate by the definition of  $\iota_\tau^-$  and  $\iota_\tau$ . Also, at any  $\sigma$ -stage,  $\iota_\sigma$  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  $s \in S_\tau$ ,  $\sigma$  has its true outcome (namely, the outcome  $o$  with  $\sigma \hat{\langle} o \rangle = \tau$ ). So for part (??), it only remains to be shown that  $\lim_{s \in S_\tau} \iota_\sigma$  exists. For this, first note that the domain of  $\iota_\sigma$  is completely determined by  $\tau$  and its guesses about tuples in  $\mathcal{A}$ .

In order to show that  $\iota_\sigma$  stabilizes at  $\tau$ -stages, note that, by induction, we are done in case  $o = \text{fin}$  (since then  $\iota_\sigma^- = \iota_\sigma$  at  $\tau$ -stages), and if  $o = (\bar{a}, n', \bar{a}')$ , we only need to show that  $\iota_\sigma(\bar{a}')$  reaches a limit over  $\tau$ -stages. But this is immediate from Lemma ?? and the description of the full strategies in section 3.1.  $\square$

Lemma ?? implies in particular that  $\iota$  is a well-defined (possibly partial) order isomorphism from  $\mathcal{A}$  into  $\mathcal{B}$ . The next lemma shows that  $\iota$  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  $\iota$ , 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  $\sigma$  has true outcome  $(a, n', \bar{a}')$ , and since  $\sigma$  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  $\tau$ -stages.

To show that  $\iota$  is surjective, fix  $b \in B$ . Fix an  $\mathcal{S}_b$ -strategy  $\sigma \subset TP$  and the tuple  $\bar{b}' \subset B$  constituting the maximal block (in order, of length  $n'$ , say) containing  $b$ . Then  $\sigma$  has true outcome  $(a, n', \bar{a}')$  (for some  $a \in A$  and  $\bar{a}' \in A^{n'}$ ) and will map  $\bar{a}'$  to  $\bar{b}'$  at all but finitely many  $\tau$ -stages.  $\square$

It is easy to see from our construction that  $\mathcal{B}$  is indeed a computable linear ordering, which by Lemma ?? 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 ??, we can fix finitely many infinite intervals  $I^0, \dots, I^k$  in  $\mathcal{A}$  such that the domain of  $\iota_\sigma^-$  consists exactly of the maximal blocks  $\bar{a}_i$  (for  $1 \leq i \leq k$ ) between these intervals. By Lemma ??, for each  $i \in [1, k]$ , we can fix  $\bar{b}_i$  such that at all but finitely many  $\sigma$ -stages,  $\iota_\sigma^-(\bar{a}_i) = \bar{b}_i$ .

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 and all in the same interval  $I^i$ . By symmetry, we'll assume that  $b <_{\mathcal{B}} f(b) <_{\mathcal{B}} f(f(b))$ .

But this contradicts the construction as described in section ??: If such  $b$  could be found, then  $\sigma$  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. □

#### 4. CONCLUDING REMARKS

In order to present a plan of attack for the full characterization of the existence of computable nontrivial self-embeddings in any computable presentation of a linear ordering, we start with a simple observation.

**Remark 4.1.** Suppose that  $\mathcal{A}$  is a countable linear ordering whose condensation is not densely ordered (and has more than one element). Then  $A$  contains an interval of order type  $\omega + 1$ ,  $1 + \omega^*$ , or  $\omega^* + \omega$ .

Since it is easy to see that any computable linear ordering containing an interval of order type  $\omega + 1$ ,  $1 + \omega^*$ , or  $\omega^* + \omega$  has a computably copy without computable nontrivial self-embedding (reference?), we can reduce our analysis for Conjecture ?? to the case where  $\mathcal{A}^*$  is a countable dense linear ordering (possibly with endpoints).

#### REFERENCES

- [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.
- [LeLN] Lempp, Steffen, *Lecture Notes on Priority Arguments*, preprint available at <http://www.math.wisc.edu/~lempp/papers/prio.pdf>.
- [So87] Soare, Robert I., *Recursively enumerable sets and degrees*, Springer-Verlag, Berlin, New York, 1987.

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 WISCONSIN, MADISON, WI  
53706-1388

*E-mail address:* kasterma@math.wisc.edu

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