On Self-Embeddings of Computable Linear Orderings ⋆

Rodney G. Downey [a], Carl Jockusch [b] and Joseph S. Miller [a]

[a] *School of Mathematical and Computing Sciences*
Victoria University, P.O. Box 600
Wellington, New Zealand

[b] *Department of Mathematics*
University of Illinois at Urbana-Champaign
1409 W. Green Street
Urbana, Illinois 61801-2975

Abstract

The Dushnik–Miller Theorem states that every infinite countable linear ordering has a nontrivial self-embedding. We examine computability-theoretical aspects of this classical theorem.

1 Introduction

The Dushnik–Miller Theorem [5] states that every infinite countable linear ordering has a nontrivial self-embedding, viz an order-preserving map from the ordering to itself that is not the identity. It is a well-known piece of folklore that this result fails to hold effectively in that there is a computable linear ordering (even of classical type ω) with no nontrivial computable self-embeddings. (See, for example, Downey [1].) Downey and Lempp [3] constructed an example of a computable linear ordering L whose classical isomorphism type is ω and such that any nontrivial self-embedding computes the halting problem. [1] They also

⋆ Research partially supported by the Marsden Fund of New Zealand. These results were established during Jockusch's visit to Wellington and Nelson as part of the NZIMA CoRE program in logic for 2003/4. Jockusch was also partially supported by the NSFC Grand International Joint Project, No. 60310213, New Directions in Theory and Applications of Models of Computation.

Email addresses: `Rod.Downey@vuw.ac.nz` (Rodney G. Downey), `jockusch@math.uiuc.edu` (Carl Jockusch), `Joe.Miller@vuw.ac.nz` (Joseph S. Miller).

[1] Downey and Lempp [3] also claim in their paper that their construction establishes that the proof-theoretical strength of the Dushnik–Miller theorem is ACA_0 in the

observed that for any computable infinite linear ordering L, $\mathbf{0}''$ has enough computational power to construct a nontrivial self-embedding of L. Later, Lempp et. al. [7] examined the question of whether $\mathbf{0}'$ is actually enough to construct a nontrivial self-embedding for a computable linear ordering. They claimed that there is a computable linear ordering with no $\mathbf{0}'$-computable nontrivial self-embedding. Unfortunately their proof contained a significant error, and hence the question remained open.

In this paper we will show in two different ways (using different kinds of orderings) that there is a computable linear ordering of ω with no nontrivial $\mathbf{0}'$-computable self-embedding. We will also show that certain computable linear orderings have nontrivial self-embeddings of degree strictly less than $\mathbf{0}''$. A linear ordering L is *discrete* if each element of L except the greatest (if any) has a successor and each element of L except the least (if any) has a predecessor. We show first that every computable discrete linear ordering of ω has a nontrivial self-embedding of degree strictly less than $\mathbf{0}''$. In fact we prove a result with a weaker hypothesis and a stronger conclusion for which we now define the terminology. If L is a linear ordering, let $S(L)$ denote the set of elements of L which have a successor. If $\mathbf{a}$ and $\mathbf{b}$ are Turing degrees, we say that $\mathbf{b}$ is *PA relative to* $\mathbf{a}$ (written $\mathbf{b} \gg \mathbf{a}$) if every $\mathbf{a}$-computable partial $\{0,1\}$-valued function can be extended to a total $\mathbf{b}$-computable function. We prove the following result:

Theorem 1 *If L is a computable linear ordering, $S(L)$ is $\mathbf{0}'$-computable and $\mathbf{b} \gg \mathbf{0}'$, then L has a $\mathbf{b}$-computable nontrivial self-embedding.*

We then show that this result is best possible in a strong sense.

Theorem 2 *There is a discrete computable linear ordering L_0 such that every nontrivial self-embedding of L_0 has degree PA over $\mathbf{0}'$ (and so strictly above $\mathbf{0}'$).*

The best possible result we might aim for would be to construct a computable linear ordering all of whose nontrivial self-embeddings compute $\mathbf{0}''$. We do not know if this is possible. We do know that no discrete linear ordering suffices for this purpose.

If L is a linear ordering, let $A(L)$ be the adjacency relation for L. Thus, the elements a, b satisfy the adjacency relation if they are distinct and no element of L lies strictly between them. The *block relation* $B(L)$ is the smallest equivalence relation on L which contains the adjacency relation. Thus the elements a, b satisfy the block relation if there are only finitely many elements of L which lie strictly between them. The equivalence classes of $B(L)$ are called

---

sense of reverse mathematics. (Here we assume that the reader is familiar with these notions.) This claim is correct, but the verification given in their paper is incomplete. We take the opportunity to include a clarification by Downey and Lempp. It is included with their permission. We put this clarification as an appendix at the end.

blocks. Clearly every block of L is either finite or order-isomorphic to one of ω, ω^* or $\omega^* + \omega$.

We might hope that if an ordering had many finite blocks it would be somehow simpler to deal with. Strangely these are the very orderings for which we seem to lack techniques. Indeed, even if an ordering has *no* infinite blocks, so that its finite condensation is isomorphic to the rationals, it seems that it still can be very complicated.

Theorem 3 *There is a computable linear ordering with no infinite blocks such that no nontrivial self-embedding is computable from* $\mathbf{0}'$.

The proof of Theorem 3 is the most technical one in this paper and is rather more unsatisfying than the earlier results. That is because it is a direct priority argument that fails to give much more insight than the fact that it is true. Nevertheless, it might admit variations which would enable more coding to happen.

Remark Dushnik and Miller [5, Theorem 1], actually proved their result in a stronger form than we have stated. They showed that every infinite countable linear ordering L has a *proper* self-embedding f, i.e. a self-embedding f whose range is not all of L. Clearly every proper self-embedding is nontrivial, but the converse fails. On the other hand, the two notions are equivalent up to Turing reducibility; for any nontrivial self-embedding f of a computable linear ordering L, there is a proper self-embedding g of L such that $g \leq_T f$. To prove this, fix $a \in L$ with $f(a) \neq a$, and assume without loss of generality that $a <_L f(a)$. Then define $g(x)$ to be $f(f(x))$ if $a \leq_L x$ and $f(x)$ otherwise. Then g is a self-embedding of L, $g \leq_T f$, and g is proper because $f(a)$ is not in the range of g. Our results are all stated for nontrivial self-embeddings but also hold for proper self-embeddings by this remark.

We prove the results on discrete orderings in Section 2, and present a $\mathbf{0}'''$ argument for Theorem 3 in Section 3. All linear orderings we consider are linear orderings of ω.

2 The discrete case

We begin with a straightforward effectivization of the original Dushnik–Miller Theorem [5, Theorem 1]. Recall that $A(L)$ is the adjacency relation and $B(L)$ is the block relation for the linear ordering L.

Proposition 4 *If L is a computable linear ordering, then L has a nontrivial self-embedding f which is Turing reducible to $A(L) \oplus B(L)$.*

Proof. The proof is essentially the same as the original proof of Dushnik and Miller. There are two cases.

Case 1. Suppose that L has an infinite block B. Fix such a block B. Then B must be either closed under successor or closed under predecessor. Without loss of generality, suppose that B is closed under successor. Define $f : \omega \to \omega$ as follows: $f(x)$ is the L-successor of x if $x \in B$, and otherwise $f(x) = x$. Then f is a nontrivial self-embedding of L and $f \leq_T A(L) \oplus B(L)$.

Case 2. Suppose that every block of L is finite. Then the finite condensation of L (i.e. $L/B(L)$) is dense. (If there were two consecutive distinct blocks, they would have to be a single block because they are finite.) Now build a nontrivial self-embedding f of L in stages, adding one new element to the domain of f at each stage. Maintain the property that if $a <_L b$, then $f(a) <_L f(b)$ and $f(a)$ and $f(b)$ lie in distinct blocks, and neither lies in the least or greatest block, if any. Also, fix $a_0 \in \omega$, and begin by defining $f(a_0)$ to be different from a_0. All of this can be done since the blocks are dense and there are infinitely many blocks. Furthermore, one obtains that $f \leq_T B(L)$. □

Note that if L is computable, then the adjacency relation $A(L)$ is Π^0_1 and the block relation $B(L)$ is Σ^0_2. It follows that L has a nontrivial self-embedding $f \leq_T 0''$, as is known. It also follows that f may be taken to be $\mathbf{0}'$-computable if $B(L)$ is $\mathbf{0}'$-computable.

In this section we will show that every discrete computable linear ordering has a nontrivial self-embedding of degree strictly less than $\mathbf{0}''$ and that there is a discrete computable linear ordering L such that every nontrivial self-embedding of L has degree strictly greater than $\mathbf{0}'$. The following notion, due to Stephen Simpson [9], will allow us to state these results in a sharp form.

Definition 5 *Let* $\mathbf{a}$ *and* $\mathbf{b}$ *be Turing degrees. Then* $\mathbf{b} \gg \mathbf{a}$ *(or* $\mathbf{b}$ *is* PA *over* $\mathbf{a}$*) means that every infinite* $\mathbf{a}$*-computable tree* $T \subseteq 2^{<\omega}$ *has an infinite* $\mathbf{b}$*-computable path.*

The following lemma is well-known. If $\mathbf{a}$ is a Turing degree, define a function f to be $\mathbf{a}$-*bounded* if there is an $\mathbf{a}$-computable function g such that $g(n) \geq f(n)$ for every $n \in \omega$.

Lemma 6 *Let* $\mathbf{a}$ *and* $\mathbf{b}$ *be Turing degrees. Then the following statements are equivalent:*

(1) $\mathbf{b} \gg \mathbf{a}$
(2) Every $\mathbf{a}$*-computable tree* $T \subseteq \omega^{<\omega}$ *which has an* $\mathbf{a}$*-bounded path has a* $\mathbf{b}$*-computable path.*
(3) Every $\mathbf{a}$*-computable* $\{0,1\}$*-valued partial function has a* $\mathbf{b}$*-computable total extension.*
(4) For any two disjoint $\mathbf{a}$*-c.e. sets* A*,* B *there is a* $\mathbf{b}$*-computable set* C *such that* C *separates* A *and* B*, i.e.* $A \subseteq C$ *and* $B \cap C = \emptyset$*.*

Furthermore, if $\mathbf{a}$ and $\mathbf{b}$ are Turing degrees, then

$$\mathbf{b} \geq \mathbf{a}' \implies \mathbf{b} \gg \mathbf{a} \implies \mathbf{b} > \mathbf{a}.$$

Here is one further well-known lemma about the relation $\gg$.

Lemma 7 *For every degree* $\mathbf{a}$ *there is an infinite* $\mathbf{a}$*-computable tree* $T \subseteq 2^{<\omega}$ *such that every infinite path through* T *has degree* $\gg \mathbf{a}$.

This lemma is easily proved from Lemma 6 using the existence of a universal $\mathbf{a}$-computable $\{0, 1\}$-valued partial function. It follows from this result and the Low Basis Theorem [6, Theorem 2.1] that for every degree $\mathbf{a}$ there is a degree $\mathbf{b} \gg \mathbf{a}$ such that $\mathbf{b}' = \mathbf{a}'$.

We do not know whether for every degree $\mathbf{b} \gg \mathbf{0}'$ every computable linear ordering of ω has a $\mathbf{b}$-computable nontrivial self-embedding. However, the next result shows that such embeddings exist for a fairly wide class of computable linear orderings.

Theorem 8 *Let* L *be a computable linear ordering of* ω *and let* $S(L)$ *be the set of numbers which have an* L*-successor. Suppose that* $S(L)$ *is* $\mathbf{0}'$*-computable. Then for all* $\mathbf{b} \gg \mathbf{0}'$*, there exists a* $\mathbf{b}$*-computable nontrivial self-embedding* f *of* L.

Proof. The following easy lemma applies to any computable linear ordering L of ω. It does not use the hypothesis that $S(L)$ is $\mathbf{0}'$-computable.

Lemma 9 *Let* L *be a computable linear ordering of* ω*. Then the following are equivalent:*

(1) For every $\mathbf{b} \gg \mathbf{0}'$ *there is a* $\mathbf{b}$*-computable nontrivial self-embedding of* L.
(2) There is a $\mathbf{0}'$*-bounded nontrivial self-embedding of* L.

Proof. We show first that (2) implies (1). Let f be a $\mathbf{0}'$-bounded nontrivial self-embedding of L, and fix $a \in \omega$ with $f(a) \neq a$. Let P be the class of all self-embeddings g of L with $g(a) \neq a$. Then P is a Π^0_1 class, so there is a computable tree $T \subseteq 2^{<\omega}$ such that P is the set of all paths through T, and furthermore, by hypothesis, there is a $\mathbf{0}'$-bounded path through T. Hence by Lemma 6 for every $\mathbf{b} \gg \mathbf{0}'$ there is a $\mathbf{b}$-computable path through T, and of course every path through T is a nontrivial self-embedding of L.

Now assume that (1) holds. By Lemma 7 there is an infinite $\mathbf{0}'$-computable tree $T \subseteq 2^{<\omega}$ such that every infinite path through T has degree $\mathbf{b} \gg \mathbf{0}'$. By the hyperimmune-free basis theorem [6, Theorem 2.4] relativized to $\mathbf{0}'$, there is a path B through T such that every B-computable function is $\mathbf{0}'$-bounded. By (1), applied to the degree $\mathbf{b}$ of B, there is a B-computable nontrivial self-embedding f of L. Then f is $\mathbf{0}'$-bounded by choice of B. □

We return now to the proof of the theorem. We are assuming now that $S(L)$ is $\mathbf{0}'$-computable and that $\mathbf{b} \gg \mathbf{0}'$. We must show that L has a $\mathbf{b}$-computable nontrivial self-embedding. By the lemma just proved, it suffices to show that L has a $\mathbf{0}'$-bounded nontrivial self-embedding. This is done by considering the following two cases.

Case 1. The ordering L has a block B which is closed under successor. Fix such a block B. Define $f : \omega \to \omega$ as in the proof of Case 1 of Proposition 4. Then f is a nontrivial self-embedding of L. Furthermore, f is $\mathbf{0}'$-bounded since $A(L)$ and $S(L)$ are $\mathbf{0}'$-computable. (Specifically, let $g(n)$ be the numerically greater of n and its L-successor if $n \in S(L)$, and otherwise let $g(n) = n$. Then g is $\mathbf{0}'$-computable, and $g(n) \geq f(n)$ (in the standard ordering of ω) for all n.)

Case 2. Assume now that Case 1 does not hold. Under this assumption, we prove that there is a $\mathbf{0}'$-computable nontrivial self-embedding of L. Note first that the block relation $B(L)$ for L is $\mathbf{0}'$-computable. To see this, observe that if $a <_L b$ then a and b lie in different blocks if and only if there is a $c \notin S(L)$ such that $a \leq c < b$. (If such a c exists, then there are infinitely many numbers between c and b, and hence there are infinitely many between a and b. If a and b lie in different blocks, then such a c must exist, since otherwise the block of a would be closed under successor, and Case 1 would apply.) This gives a Π^0_2 definition of the block relation, and as mentioned the block relation is Σ^0_2 in every computable linear ordering. It now follows from the remarks after the proof of Proposition 4 that there is a $\mathbf{0}'$-computable nontrivial self-embedding of L. □

Corollary 10 *Suppose that L is a computable discrete linear ordering. Then L has a nontrivial self-embedding of degree $\mathbf{b} < \mathbf{0}''$, in fact with $\mathbf{b}' \leq \mathbf{0}''$.*

The following is a corollary to the proof of Theorem 8.

Corollary 11 *Suppose that L is a computable linear ordering and that $0'' \not\leq_T 0' \oplus S(L)$. Then there is a function f such that $0'' \not\leq_T f$ and f is a nontrivial self-embedding of L.*

Proof. Let $C = 0' \oplus S(L)$. Suppose first that L has a block closed under successor, so that Case 1 in the proof of Theorem 8 applies. By the discussion in Case 1, there is a C-bounded nontrivial self-embedding f of L. Fix $a \in \omega$ with $f(a) \neq a$, and let P be the class of order-preserving maps g from L into L with $g(a) \neq a$. Then P is a Π^0_1 subset of ω^ω with a C-bounded element, so it follows from [6, Theorem 2.5] relativized to C, that P has an element f which does not compute $0''$. If Case 2 in the proof of Theorem 8 applies, we show that L has a C-computable nontrivial self-embedding, which suffices since $0'' \not\leq_T C$ by hypothesis. By Proposition 4 it suffices to show that $A(L) \oplus B(L) \leq_T C$. Clearly, $A(L) \leq_T 0' \leq_T C$. Also $B(L) \in \Sigma^0_2 = \Sigma^{0,0'}_1 \subseteq \Sigma^{0,C}_1$. Finally $\overline{B(L)} \in \Sigma^{0,S(L)}_1 \subseteq \Sigma^{0,C}_1$ by the proof of Case 2 of Theorem 8, so $B(L) \leq_T C$, as needed to complete the proof. (The authors thank Joseph Mileti for pointing out an error in their original proof of this corollary and for supplying this corrected proof.) □

We now show that Theorem 8 is optimal in a certain strong sense.

Theorem 12 *There is a computable discrete linear ordering L of ω such that every nontrivial self-embedding of L has degree $\gg \mathbf{0}'$. (In particular $S(L)$ is* *computable, so by Theorem 8 the degrees* $\mathbf{b}$ *which compute nontrivial self-embeddings of* L *are precisely the degrees* $\mathbf{b} \gg \mathbf{0}'$*.)*

Proof. We first need another standard lemma.

Lemma 13 *Let* $\mathbf{a}$ *be a Turing degree. Then there are disjoint* $\mathbf{a}$*-c.e. sets* A_0*,* A_1 *such that every set which separates* A_0 *and* A_1 *has degree* $\gg \mathbf{a}$*.*

Proof. Let θ be a universal $\mathbf{a}$-computable partial $\{0, 1\}$-valued function, and let $A_i = \theta^{-1}(i)$ for $i = 0, 1$. Apply Lemma 6. □

By the previous lemma (with $\mathbf{a} = \mathbf{0}'$) there exist disjoint Σ^0_2 sets A_0, A_1 such that every set C which separates A_0, A_1 is of degree $\gg \mathbf{0}'$. Since each A_i is Σ^0_2 there exist uniformly computable sets $A_{i,s}$ for $i \leq 1, s \in \omega$ such that, for all x and for $i \leq 1$,

$$x \in A_i \text{ if and only if } (\forall^\infty s)[x \in A_{i,s}].$$

(Here $(\forall^\infty s)$ means "for all but finitely many s".) To obtain the computable approximations $A_{i,s}$, let h be a computable function such that for all $a \in \omega$ and $i \leq 1$,

$$a \in A_i \text{ if and only if } W_{h(a,i)} \text{ is finite.}$$

Then define

$$A_{i,s} = \{a : W_{h(a,i),s} = W_{h(a,i),s+1}\}.$$

We will define a computable linear ordering L. First we give an intuitive sketch of the construction. Let f be a nontrivial self-embedding of L, and suppose that $f(a_0) \neq a_0$. We give a strategy (independent of f and a_0) for defining L so as to show that the degree of f is $\gg \mathbf{0}'$. Suppose for the moment that $a_0 <_L f(a_0)$. We will let 0 be the least element for L and 1 be the greatest element. By replacing a_0 by $f(a_0)$ if necessary, we may assume that $a_0 \neq 0$. Define

$$S = \{b : b <_L a_0 \text{ or } b <_L f(b)\}.$$

Note that $0 \in S$ and $1 \notin S$. The following simple lemma is crucial.

Lemma 14 *If* $b \in S$*,* $b <_L c$*, and the* L*-interval* $[b, c]$ *is finite, then* $c \in S$*.*

Proof. It suffices to show that S is closed under successor, so suppose for a contradiction that $b \in S$, c is the successor of b, and $c \notin S$. If $b <_L f(b)$, then $c \leq_L f(b) <_L f(c)$, so $c \in S$, a contradiction. Since $b \in S$ and it is not the case that $b <_L f(b)$, it follows that $b <_L a_0$. Hence $c \leq_L a_0$, as c is the successor of b. Since $c \notin S$, we have also that $c \geq_L a_0$, so $c = a_0$. But $a_0 \in S$ since $f(a_0) > a_0$, and hence $c \in S$, which is the desired contradiction. □

Note that S is f-computable, so it suffices to show that there is a set $C \leq_T S$ such that C separates the disjoint Σ^0_2 sets A_0 and A_1 described above. To do this, we first choose a number $w_0 \neq 0, 1$ and decree that $0 <_L w_0 <_L 1$. We use the computable approximations $A_{i,s}$ to ensure that the interval $[0, w_0]$ is finite if $0 \in A_0$, and the interval $[w_0, 1]$ is finite if $0 \in A_1$. Since $0 \notin A_0 \cap A_1$, this is compatible with making the field of $<_L$ infinite (in fact all of ω).

Suppose that $0 \in A_0$. Then $[0, w_0]$ is finite by construction, and also $0 \in S$, as already mentioned. It follows from Lemma 14 that $w_0 \in S$. One may show by a parallel argument that if $0 \in A_1$, then $w_0 \notin S$. Thus, in constructing an S-computable set C which separates A_0 and A_1, it is safe to put 0 into C if and only if $w_0 \in S$. We now bisect the interval $[0, 1]$ into the subintervals $[0, w_0]$ and $[w_0, 1]$. One of these subintervals, say $[a, b]$, has the property that $a \in S$ if and only if $b \notin S$. Thus, to ensure that $C(1)$ is computable from S, we can repeat the same process, but starting with $[a, b]$ in place of $[0, 1]$. That is, we pick a witness w_1 in the interval (a, b) and ensure that $[a, w_1]$ is finite if $1 \in A_0$ and that $[w_1, b]$ is finite if $1 \in A_1$. We then put 1 into C if and only if $w_1 \in S$. This bisection process is then iterated in the obvious way to code a separating set C into S.

One small difficulty with the above argument is that L must be computable and not merely S-computable. Unfortunately, there is no reason to think that we can compute effectively which subinterval of a given interval to use. The solution is simple. We allow the construction to use both subintervals. For example, there is only one "version" of w_0, but there may be two "versions" of w_1, one in the subinterval $(0, w_0)$ and the other in the subinterval $(w_0, 1)$. In general, there may be up to 2^n "versions" of w_n, but S can compute which is the correct version, w_n^*, and we let $C = \{n : w_n^* \in S\}$.

Given the above intuitive description, it is straightforward to work out the details, and the reader may wish to do this as an exercise. However, a construction and verification are included below for completeness.

In the construction, we define a computable partial function θ which associates with certain intervals $[a, b]$ a c.e. set $W_{\theta(a,b)}$. When we do this, we are requiring that the interval $[a, b]$ be finite if $W_{\theta(a,b)}$ is finite. Let R_s denote the set of elements which have been put into the field of L by the end of stage s. $\theta(a, b)$ will be defined if and only if there is a stage s such that a and b are L-consecutive elements of R_s. We also define a computable function r. When we set $r(x) = n$, we are letting x play the role of some version of w_n in the intuitive sketch.

Stage 0. Let $R_0 = \{0, 1\}$ and decree that $0 <_L 1$. Choose $\theta(0, 1)$ so that $W_{\theta(0,1)} = \omega$. Let $r(0) = r(1) = -1$.

Stage $s + 1$. Call an interval $[a, b]$ *receptive* at stage $s + 1$ if

(1) $a, b \in R_s$ and $a <_L b$
(2) There is no $c \in R_s$ such that $a <_L c <_L b$
(3) For every interval $[a', b']$ with $a', b' \in R_s$, $a' \leq_L a <_L b \leq_L b'$, $\theta(a', b')$ already defined, and $\langle a', b'\rangle \leq \langle a, b\rangle$, it is the case that $|W_{\theta(a',b'),s}| \geq \langle a, b\rangle$.

(Condition (3) above says roughly that putting a new element in the interval $[a, b]$ would not be too injurious to commitments made that various intervals must be finite.) If no interval is receptive at stage $s + 1$, do nothing, i.e. let $R_{s+1} = R_s$ and leave θ and r unchanged. Otherwise, fix the least $\langle a, b\rangle$ such that the interval $[a, b]$ is receptive. Let c be the least number not in R_s. Put $R_{s+1} = R_s \cup \{c\}$ and decree that $a <_L c <_L b$, thus determining L on R_{s+1} via transitivity. Define

$$r(c) = \max\{r(a), r(b)\} + 1.$$

where of course the maximum is taken over the standard ordering of ω. Finally, choose $\theta(a, c)$ and $\theta(c, b)$ so that

$$W_{\theta(a,c)} = \{s : r(c) \notin A_{0,s}\} \quad \text{and} \quad W_{\theta(c,b)} = \{s : r(c) \notin A_{1,s}\}.$$

We say that the interval $[a, b]$ *receives attention* at stage $s+1$. This completes the construction.

Lemma 15 *$\bigcup_s R_s = \omega$.*

Proof. First, show by induction on s that there exist $a, b \in R_s$ such that a and b are L-consecutive elements of R_s and for all $a', b' \in R_s$ such that $a' \leq_L a <_L b \leq_L b'$ with $\theta(a', b')$ defined, the set $W_{\theta(a',b')}$ is infinite. This is obvious for $s = 0$. Now assume that it is true at s, and fix corresponding values of a and b. It clearly holds at $s + 1$ (with the same choice of a and b) except possibly if a new element c is inserted between a and b at stage $s + 1$. The statement then holds at $s + 1$ via the pair a, c if $W_{\theta(a,c)}$ is infinite, so it holds if $r(c) \notin A_0$. Similarly, it holds via the pair c, b if $r(c) \notin A_1$. Since A_0 and A_1 are disjoint, the statement holds at $s + 1$.

Assume now for a contradiction that $\bigcup_s R_s \neq \omega$. Since numbers enter $\bigcup_s R_s$ in natural order, it follows that $\bigcup_s R_s$ is finite. Fix s so that $R_t = R_s$ for all $t \geq s$. Choose $a, b \in R_s$ which satisfy the conditions stated in the previous paragraph. Then the interval $[a, b]$ is receptive at all sufficiently late stages. Hence there exists $t > s$ with $R_t \neq R_s$, which gives us the desired contradiction. □

It follows from the above lemma and the effectiveness of the construction that L is a computable linear ordering of ω. The next lemma shows that the required intervals are finite.

Lemma 16 *If $\theta(a', b') \downarrow$ and $W_{\theta(a',b')}$ is finite, then the interval $[a', b']$ is also finite.*

Proof. Let $|W_{\theta(a',b')}| = k$. If a new element is put into $[a', b']$ at stage $s+1$ and $\theta(a', b')$ is defined by stage $s + 1$, then the numbers a, b chosen at stage $s + 1$ satisfy either $\langle a, b\rangle \leq \langle a', b'\rangle$ or $\langle a, b\rangle \leq k$. There are only finitely many such pairs (a, b), and each receives attention at most once, so the interval $[a', b']$ is finite. □

We now define an S-computable sequence of intervals $[a_n, b_n]$ by recursion. The intervals will have the following properties:

(1) $a_n \in S$ and $b_n \notin S$
(2) $\theta(a_n, b_n) \downarrow$
(3) For all $\langle a', b' \rangle$ with $a' \leq_L a_n <_L b_n \leq_L b'$ and $\theta(a', b') \downarrow$ the set $W_{\theta(a',b')}$ is infinite

Let $[a_0, b_0] = [0, 1]$. Now suppose that $[a_n, b_n]$ is defined and satisfies the above conditions. If the interval (a_n, b_n) is empty, then the interval $[a_n, b_n]$ is receptive at all sufficiently late stages and hence the open interval (a_n, b_n) is nonempty, a contradiction. (There are only finitely many intervals $[a, b]$ with $\langle a, b \rangle < \langle a_n, b_n \rangle$, and each receives attention at most once.) Let w_n be the first element placed into the interval $[a_n, b_n]$. If $w_n \notin S$, let $[a_{n+1}, b_{n+1}] = [a_n, w_n]$, and otherwise let $[a_{n+1}, b_{n+1}] = [w_n, b_n]$. Condition (1) clearly holds for $[a_{n+1}, b_{n+1}]$, and (2) holds by construction. To prove (3) for $[a_{n+1}, b_{n+1}]$ it suffices (using that (3) holds of $[a, b]$) to show that $W_{\theta(a_{n+1},b_{n+1})}$ is infinite. Suppose first that $w_n \notin S$. Then $a_{n+1} = a_n \in S$ by inductive hypothesis. It follows from Lemma 14 that the interval $[a_{n+1}, b_{n+1}]$ is infinite. Hence, by Lemma 16, $W_{\theta(a_{n+1},b_{n+1})}$ is infinite. The case where $w_n \in S$ is analogous.

Let $\{[a_n, b_n]\}_{n\in\omega}$ be the S-computable sequence of intervals defined above and, as above, let w_n be the first element placed in the open interval (a_n, b_n). Of course, the sequence $\{w_n\}_{n\in\omega}$ is also S-computable. Furthermore, it is easy to check by induction on n that $r(w_n) = n$ for all $n \in \omega$. (One should include in the induction the fact that $r(a_n), r(b_n) < n$ and use the fact that either $a_{n+1} = w_n$ or $b_{n+1} = w_n$.)

Suppose now that $n \in A_0$. Hence $W_{\theta(a_n,w_n)} = \{s : n \notin A_{0,s}\}$ is finite. Hence by Lemma 16 the interval $[a_n, w_n]$ is finite. As $a_n \in S$, it follows from Lemma 14 that $w_n \in S$. Similarly, if $n \in A_1$, it follows that $w_n \notin S$. Let

$$C = \{n : w_n \in S\}.$$

It follows from the above remarks that C separates A_0 and A_1. Since $C \leq_T S \leq_T f$ (where f is the given nontrivial self-embedding of L), it now follows from the choice of A_0 and A_1 that the degree of f is PA over $\mathbf{0}'$.

In the above proof it was assumed that $a_0 <_L f(a_0)$. If there is no such a_0, instead choose an element $a_1 \neq 1$ with $f(a_1) <_L a_1$. The verification works in much the same way, but with L replaced by its dual ordering. For instance, S would be defined as $\{b : a_1 <_L b \text{ or } f(b) <_L b\}$. Of course, it is important that the construction remains exactly the same and only the verification changes. Further checking of this case is left to the reader.

It remains to verify that the ordering L is discrete. Suppose for a contradiction that $d \neq 1$ and d has no successor under L. It follows that infinitely often d plays the role of a in the construction. Fix s_0 with $d \in R_{s_0}$. For $s \geq s_0$ let

$h(s) = r(b)$, where b is the L-least element of R_s with $a <_L b$. By construction, we have $h(s+1) \geq h(s)$ for all $s \geq s_0$. Furthermore, $h(s+1) > h(s)$ if d plays the role of a at stage $s+1$. Since there are infinitely many such s, h is unbounded. Choose $s_1 \geq s_0$ with $h(s_1) > r(d)$. Then, for all $s \geq s_1$, either $h(s+1) = h(s)$ or $h(s+1) = h(s)+1$. Since the latter occurs infinitely often, the range of h is cofinite. Choose s such that $h(s+1) = h(s)+1 \in A_0$. At stage $s+1$, for some $c >_L d$ one sets $W_{\theta(d,c)} = \{s : r(c) \notin A_{0,s}\}$, where $r(c) = h(s+1) \in A_0$. Hence $W_{\theta(d,c)}$ is finite, and thus by Lemma 16, the interval $[d,c]$ is finite. This contradicts the assumption that d has no successor. The proof that every number except 0 has a predecessor is analogous. $\square$

Remark The order type of the ordering L constructed in Theorem 12 is $\omega + \zeta \cdot \eta + \omega^*$, where ζ is the order-type of the integers and η is the order-type of the rational numbers. Since L is a discrete linear ordering with first and last elements, to see this it suffices to show that the blocks of L are densely ordered. Suppose that $u <_L v$ and u, v belong to different blocks. Fix s_0 so that $u, v \in R_{s_0}$. We now define recursively a sequence of intervals $[a_n, b_n]$ with the following properties:

(1) $u \leq_L a_n <_L b_n \leq_L v$
(2) $[a_n, b_n]$ is infinite
(3) $\theta(a_n, b_n) \downarrow$

Choose $[a_0, b_0]$ so that a_0 and b_0 are L-consecutive elements of R_{s_0} with $u \leq_L a_0 <_L b_0 \leq_L v$ and $[a_0, b_0]$ infinite. Such a_0, b_0 exist because $[u, v]$ is infinite. Clearly (1)–(3) with $n = 0$ hold for $[a_0, b_0]$. Let $[a_n, b_n]$ satisfying (1)–(3) be given. Let w_n be the first element put into the interval $[a_n, b_n]$ after $\theta(a_n, b_n)$ becomes defined. If the interval $[a_n, w_n]$ is infinite, let $[a_{n+1}, b_{n+1}]$ be this interval, and otherwise let $[a_{n+1}, b_{n+1}] = [w_n, b_n]$. Let $h(n) = r(w_n)$, where r is defined as in the proof of Theorem 12. Argue as in the final paragraph of the proof of Theorem 12 that the range of h is cofinite. Since $A_0 \cup A_1$ is coinfinite, there exists n such that $h(n) \notin A_0 \cup A_1$. Hence $W_{\theta(a_n, w_n)}$ and $W_{\theta(w_n, b_n)}$ are both infinite. From this it can be shown that the intervals $[a_n, w_n]$ and $[w_n, b_n]$ are both infinite. (We omit the easy details.) Hence the block of w_n lies strictly between the blocks of u and v, and this completes the proof that the blocks are densely ordered.

3 The case with no infinite blocks

In this section we will prove Theorem 3. That is, we will construct a computable linear ordering with no infinite blocks and which has no nontrivial self-embedding computable from $\mathbf{0}'$. This is a direct priority argument obtained by modifying the incorrect argument of Lempp, McCoy, Morozov, and Solomon [7]. Let $\{\varphi_e(\cdot,\cdot) : e \in \omega\}$ denote an enumeration of all primitive recursive functions of two variables. We must build the ordering $A = \langle A, <_A \rangle$ to satisfy the requirement

$$R_e : \text{ If } \forall x \lim_s \varphi_e(x,s) \text{ exists, then } \lim_s \varphi_e(\cdot,s)$$
$$\text{is not a nontrivial self-embedding of } A.$$

In the construction to follow, we will assume that if we have an α-stage s where $\varphi_e(x,\cdot)$ has changed its value since the last α-stage *then we will regard* $\varphi_e(x,s)$ *to have the new value* (as opposed to letting it change back to the old value and risk missing the change). This convention saves considerably on notation.

As well as the R_e requirements, we must ensure that the ordering we build has only finite blocks. For each $z \in A$ we will keep track of the minimal block around z which respects the intervals preserved by higher priority worker nodes (belonging to R_e requirements as described below); denote this at stage s of the construction as $B_z[s]$. This gives rise to the negative requirement

$$N_z : \lim_s B_z[s] = B_z \text{ exists.}$$

The basic module for the N_z is pretty obvious. At each stage s, we will have two points, $l(z,s)$ and $r(z,s)$, the left and right points of $B_z[s]$. Then at the very least, we will turn these into right and left limit points respectively by adding a new point immediately to the left of $l(z,s)$ and another immediately to the right of $r(z,s)$. It is making such a strategy live with the one we use for the R_e-type requirements which creates the difficulty. As we now see, the R_e-type requirements tend to want us to preserve various parts of the ordering as blocks, directly in conflict with the N_z.

We now turn to the satisfaction of the R_e. The R_e are quite difficult to handle as a single entity, and hence we will decompose them into subrequirements of the form:

$$R_{e,x,m,n} : \text{ If } \lim_s \varphi_e(x,s) = m \neq x \wedge \lim_s \varphi_e(m,s) = n \neq m,$$
$$\text{then } \lim_s \varphi_e(\cdot,s) \text{ is not a self-embedding.}$$

Basic Module. The basic module is pretty straightforward. Fix e, x, m, n. We need worry only if $x <_A m <_A n$, or $n <_A m <_A x$, since otherwise if the hypothesis of the requirement is satisfied, then $\lim_s \varphi_e(\cdot,s)$ cannot be isotone. Thus, without loss of generality, we suppose that $x <_A m <_A n$. The basic module is to perform the following steps.

(i) Wait for $\varphi_e(x,s) = m$, and $\varphi_e(m,s) = n$.
(ii) Impose a restraint $r(e,x,m,n,s)$ which wants to preserve the interval $[m,n]$, so that $|[m,n][s]| = |[m,n]|$. Add $|[m,n][s]|$ many new points between x and m.
(iii) At stages $s' > s$, if you see either $\varphi_e(x,s') \neq m$, or $\varphi_e(m,s') \neq n$, drop all restraint and go back to (i).

Outcomes. The basic module has the following outcomes in order of priority, starting with the weakest.

(a) f. We wait for (i) above for almost all stages. This is a *finite* outcome.
(b) d. We implement (ii) at some stage and never initialize this action. This is a *global* win for R_e and is a *finite* outcome. This outcome will ensure that all $y \in [m, n]$ will be tied together in the same finite block.
(c) ∞. This outcome has stronger priority than those above, and is the outcome that we cycle through (i) and (ii) infinitely often. This is the dangerous outcome in that its action will be to put infinitely many points between x and m. We must be rather careful to make sure that they are not amalgamated into a single block. Additionally, we must be careful to make sure that we can preserve other things, as we see below. Note that the ∞ outcome is a *global* win for R_e.

Coherence. We discuss the coherence of the various requirements. We will assume that the reader is familiar with the methodology of *linking* as per the monograph of Soare [10] in this discussion. The R_e will have a single *top* or *mother* node τ and the $R_{e,x,m,n}$ will have one or more versions at nodes σ along each path in the cone below τ. As usual, if $e < f$, then since the global priority of R_e is higher than that of R_f the mother node τ_e for R_e will be above that for R_f, namely τ_f.

Now we begin by examining the relationship between two σ nodes, say σ_1 devoted to satisfying $R_{e,x,m,n}$ and σ_2 devoted to $R_{f,y,p,q}$, with R_e of higher global priority than R_f. First we suppose that $\sigma_1 \subset \sigma_2$.

The first possibility is that $\sigma_1\hat{\ }f \subseteq \sigma_2$. This case is straightforward. $\sigma_1\hat{\ }j$ for $j \neq f$ will initialize σ_2 each time they are played, but this will happen only finitely often. After that σ_1 has *no* effect on σ_2.

The second possibility is that $\sigma_1\hat{\ }d \subseteq \sigma_2$. In this case, d represents a global win for e, and hence σ_2's mother node τ_f will be between $\sigma_1\hat{\ }d$ and σ_2. Again this is all finitary and σ_1 has finite effect on τ_f. That is, τ_f will *know* that σ_1 has restrained $[m, n]$ from some point onwards. The effect that this has on the nodes below τ_f is that σ_2 will not be able to increase $|[y, p]|$ should $[y, p] \subset [m, n]$. We could use the option of "deleting" all such requirements (or making them inactive). After all, any nontrivial self embedding must move infinitely many numbers. But too much deletion would get us into trouble, so we actually use the "h" outcome which will be described below. This is the option that the hypothesis is correct; instead of deleting the requirement, this outcome replaces it with one compatible with the restraints above it. More on this later.

The final possibility is that $\sigma_1\hat{\ }\infty \subseteq \sigma_2$. Again we will ensure that σ_2's mother node is $\sigma_1\hat{\ }\infty \subseteq \tau_f \subset \sigma_2$. Now it is certainly true that σ_2 will only be accessible at stages where we have just cycled through (ii) of the basic module, but that something has changed so that (iii) is invoked.

The problem is the following. Each time we invoke $R_{e,x,m,n}$ at σ_1, we will increase the cardinality of the interval $[x,m]$. Note that τ_f and hence σ_2 will of course know that $|[x,m]| \to \infty$. In particular, nodes in the cone below $\sigma_1\widehat{\ }\infty$ will know not to even attempt to preserve an interval $[p,q] \supseteq [x,m]$ nor $[q,p] \supseteq [x,m]$. This is because no such preservation could hope to succeed. Indeed, if the union of the intervals preserved by a finite collection of nodes were to cover $[x,m]$, then at least one of them would fail to succeed.

However, notice that *should* there be a nontrivial self-embedding with $\varphi_f(y) = p$ and $\varphi_f(p) = q$, then φ_f must also move q. This means that, if for instance $[p,q] \supseteq [x,m]$, then we can still diagonalize against φ_f while avoiding the interval $[x,m]$.

Additionally, notice that *we* control where the new points are put into $[x,m]$. Currently we are treating the action of $R_{e,x,m,n}$ as if it was immaterial where new points were placed. Our idea will be to place the points as close as possible to x, namely make the apparent end points of $B_x[s]$ limit points.[2] Thus nodes of lower global priority, such as σ_2 below $\sigma_1\widehat{\ }\infty$, will essentially be free to restrain their $[p,q]$ should we see $\varphi_f(y) = p$ unless $p \leq r(x)$, the right hand end point of B_x, and $q > r(x)$. Again we note that should φ_f be a self-embedding, then it would by necessity move, for instance, q in such circumstances.

Again we might choose to delete the offending requirement from the cone below $\sigma_1\widehat{\ }\infty$, but once again we use the "h" outcome described below. The problematical requirements which want to preserve something not preservable will be replaced by "safe" requirements. Note that this is a different use of the "h" outcome than above, where we used it to replace requirements which wanted to add points to preserved intervals. Again we defer our description of this outcome.

This brings us to the case where now $\sigma_2 \subset \sigma_1$.

The first case we consider is where $\sigma_2\widehat{\ }\infty \subseteq \sigma_1$. Now whilst the local priority of σ_2 is higher than that of σ_1, the global priorities are reversed. This situation will of course entail

$$\tau_e \subset \tau_f \subset \sigma_2\widehat{\ }\infty \subseteq \sigma_1.$$

Although $[y,p]$ looks like it will be infinite, we will still need to consider $R_{e,x,m,n}$ with, say, $B_y \cap [m,n] \neq \emptyset$.

We would *like* to be able to preserve $[m,n]$. It is here that we note that at any occasion that we preserve $[m,n]$ we have a potential *global win* for R_e. *Whilst this situation occurs we have the potential to be a new strategy for the sake of* R_f.

[2] The astute reader will note that our action at $R_{e,x,m,n}$ is very much like the action of N_x.

Thus our solution will be a typical $\mathbf{0}'''$ strategy. At the very stage where σ_1 is accessible so that $\varphi_e(x, s) = m$ and $\varphi_e(m, s) = n$, we will *link* from τ_e to σ_1. That is, at stages s' after s, until $\varphi_e(x, s') \neq m$ or $\varphi_e(m, s') \neq n$, when we hit the mother node τ_e we will travel the link (τ_e, σ_1) skipping over the intermediate nodes. Notice that when we hit σ_1 we will play $\sigma_1\hat{\ }d$, until either $\varphi_e(x, s') \neq m$ or $\varphi_e(m, s') \neq n$.

Of course, should we eventually see $\varphi_e(x, s') \neq m$ or $\varphi_e(m, s') \neq n$, then we will travel the link one last time to play $\sigma_1\hat{\ }\infty$. After that, we will remove the link. We will then be able to visit σ_2 again, if necessary. Notice that σ_2 does not care about the linking whilst the link is there, since basically at that stage σ_2's hypothesis seems wrong, and hence it imposes no constraints on the construction. If σ_2 is actually on the true path, what will happen is that it will be awoken from its slumber (it sleeps whilst it is being linked over) infinitely often, and its actions will be met.

The conclusion is that *either* there is some σ_1 in the cone below $\sigma_2\hat{\ }\infty$ which is linked to for *almost all* stages, in which case we have globally won R_e with *finite* effect (namely preserving $[m, n]$ from some s onwards), *or* every link created is later removed. This means that $\sigma_2\hat{\ }\infty$ will be on the *genuine true path* (GTP) which is the leftmost collection of nodes which are actually visited infinitely often.[3] All that will happen is that there will be many *delays* in the action of σ_2. This causes no real grief.

The next case we need to consider is where $\sigma_2\hat{\ }d \subseteq \sigma_1$. Strangely enough, this is the most complex situation. This case will be played when we have seen $\varphi_f(y, s) = p$ and $\varphi_f(p, s) = q$. As usual, we assume that $y <_A p <_A q$. We will have increased the cardinality of $[y, p]$ and want to preserve $[p, q]$. Assuming that, without loss of generality, $x <_A m$, the problematical case will clearly be if $[x, m] \subseteq [p, q]$. This is because σ_1 would desire to increase the cardinality of $[x, m]$, yet $[p, q]$ wishes to preserve it.

The critical observation is that σ_1 will know that the cardinality of $[p, q]$ will be finite. Thus, letting B_q denote the complete block of $[p, q]$, any self-embedding that moves x, must move (in the situation above) $r(q)$, the rightmost point of B_q. (In the situation where $m <_A x$, then we would use $l(q)$.) This is a device from the flawed proof of Lempp et. al. [7].

The device we use here is to insert a new outcome h (for "hypothesis correct") to σ_1, between d and f. In the cone below $\sigma_1\hat{\ }h$, we will only have R_e worker requirements σ_1' of the form $R_{e,r(q),m',n'}$ (assuming that $x <_A m$) for $r(q) <_A m' <_A n'$.

The reason we can do this is because we "know" that if φ_e really is a nontrivial self-embedding it *must* move $r(q)$. Thus, we should be able to diagonalize using

[3] This is a notion from Downey and Stob [4], which has found a number of applications, such as Downey, LaForte and Shore [2].

$r(q)$ as a starting point.

In the actual construction, it seems easiest[4] to do this by introducing new sub-outcomes of the outcome h. These are of the form $\langle h, x, m, k\rangle$, and $\langle h, m, x, k\rangle$. The first says that "hypothesis correct, $x <_A m$ and the block that σ_1 must respect has right end point k" (In the notation above, $r(q) = k$.) The other one says that $m <_A x$, and the analogous thing about the left end point. Naturally, we have $\langle h, m, x, k\rangle$ left of $\langle h, m, x, k'\rangle$ if $k > k'$.

Now the reader might ask, "why can't this situation re-occur infinitely often and hence again we would fail to meet R_e?".

But now our situation is quite different. With priority σ_1 we will protect B_q. Thus $r(q)$ is a left limit point. Hence no *requirement $R_{g,w,k,l}$ can try to preserve $[r(q), v]$ for any v.* For g of arbitrary global priority, the usual argument will hold. That is, if α is a worker node for R_g, and $[k, l]$ (or $[l, k]$ as the case might be) intersects B_q, then φ_g, if it is a self-embedding, must also move $r(q)$ (or $l(q)$). Thus, again we play the "h" outcome for such requirements in the cone below $\sigma_1\frown h$ and they cause no problems.

Notice to be thematic, we will also be able to use linking here. If we see σ_1' extending $\sigma_1\frown h$ looking accessible and its hypothesis looking correct, we could play $\sigma_1'\frown d$, and perhaps later $\sigma_1'\frown\infty$ linking directly from τ_e. This is because the action is now arranged to cause no injury to $\sigma_2\frown d$.

Now this, in turn, creates a couple more problems. Suppose we play some h-type outcome, say $\langle h, x, m, k\rangle$, the next time that we hit the node σ_1, we might well see that the hypothesis is incorrect again, in that $\varphi_e(x) \neq m[s']$, or $\varphi_e(m) \neq n[s']$. At such a stage s', we would like to play an infinite outcome, like ∞, of σ_1 to draw attention to the fact that we might be able to witness a global win for R_e at σ_1, were this to re-occur infinitely often.[5] Moreover, the action at σ_1 when we play $\sigma_1\frown\langle h, x, m, k\rangle$ is to enumerate points on each side of some block determined at σ_1 by k. This has a similar effect on nodes trying to preserve a block including k as the d outcome of σ_1. Its effect is like an N_z node, and needs two parameters, k_1 and k_2, the left and right sides of the relevant block. Our solution is to add infinitely many new outcomes of the form $\langle k_1, k_2, \infty\rangle$ between the outcomes of the form $\langle h, \cdot, \cdot, \cdot\rangle$ and the d outcome. Note that nodes extending $\langle k_1, k_2, \infty\rangle$ will know that k_1 and k_2 are the left and right end points of a block, and hence will know that this will need to be treated exactly as an N_z block, as described below.

[4] We could also have put a new N_q node immediately below $\sigma_1\frown h$, but then we would need to dynamically decide which nodes in the cone to inactivate.

[5] Strangely enough, this extra outcome can be avoided, but the proof becomes more complex, since there is no actual outcome witnessing that φ_e is not real. We have chosen the current method as the verification and construction become more perspicuous.

Remark Actually, with a little more complexity, we could also have used the ∞ outcome of σ_1 in place of the $\langle k_1, k_2, \infty\rangle$ outcomes. We believe the present method is slightly simpler.

N_z vs R_e. Now we turn to the coherence of the requirements N_z and the diagonalization ones, the R_e. We will be representing the action of N_z by a node ν on the tree. This will have outcomes $\ldots, 3, 2, 1$, of order type ω^*. The idea is that these outcomes represent the eventual cardinality of the complete block around z, B_z, in the ordering A.

The basic strategy is above. We will make the end points $l(z)$ and $r(z)$ limit points as best we can, by adding appropriate new points at each stage. A node $\sigma_1\hat{\ }d \subseteq \nu$ will be able to preserve some $[m, n]$ and hence could well make $B_{z(\nu)}$ bigger. However, if $\sigma_1\hat{\ }d$ is on the true path, its effect will be finite. The same story holds for $\sigma_1\hat{\ }h \subseteq \nu$.

The case where things might go awry is where $\sigma_1\hat{\ }\infty \subseteq \nu$. Here $[x, m]$ is being made infinite. This might potentially make the size of B_z infinite. If ν is below $\sigma_1\hat{\ }\infty$, then at the stage where ν is being accessed, σ_1 does not care what happens. As far as σ_1 is concerned, $|B_z|$ can be 1. Thus, if ν was immediately below $\sigma_1\hat{\ }\infty$ at the stage it was accessed, we could return $|B_z|$ back to 1. Thus this outcome has essentially no effect on ν.

A finite number of σ_i' above again will cause no problem since the only really important cases are the d outcomes and these are finitary.

Finally we consider the case where we have a node ν above a σ_1. If ν has higher global priority than σ_1 so that σ_1's mother is between some outcome of ν and σ_1, then ν will have finite effect upon R_e in that it might cause some of the $R_{e,x,m,n}$ requirements to follow an "h" outcome. Again this is because $B_{\nu(z)}$ will force $l(z)$ (resp. $r(z)$) to have no predecessor (resp. successor). Thus τ_e and hence σ_1 will treat it exactly like an infinite outcome of some higher priority $\sigma_2\hat{\ }\infty$. Note that ν will have only finite *global* effect.

Hence, the only problematical case will of course be if σ_1's mother node is above ν. Suppose that $\tau_e \subset \nu\hat{\ }i \subseteq \sigma_1$. Again local/global considerations come to our rescue. Should σ_1 of higher global priority than ν look correct and, say, $m <_A l(z) \leq_A n$, then we will play the outcome $\sigma_1\hat{\ }d$, whilst these conditions remain in force, and as before, create a link (τ_e, σ_1). This link will not matter to ν should it later be removed since before we pass ν again we would be acting to make $l(z)$ and $r(z)$ limit points. They do not care about the delay created by the link. On the other hand, if the link is there for almost all stages, then again we would have a global win for R_e, and with finite effect on ν, which would eventually be met by the backup strategy guessing $\sigma_1\hat{\ }d$. Of course, $\sigma_1\hat{\ }d$ has only finite effect on this backup strategy.

We now turn to the formal details.

3.1 The priority tree

Our priority tree will have nodes of three types. The first type are *mother* nodes τ, which have a single outcome o, and will be assigned to some global requirement R_e, so that we will write $e = e(\tau)$. These are nodes of length $\equiv 0 \pmod 3$. The next type are *worker* nodes σ which are devoted to subrequirements of some R_e, and hence will be assigned to some e, x, m, n. These are assigned to nodes of length $\equiv 1 \pmod 3$. We form the tree so that such σ occur below some τ with $e(\tau) = e$. For the longest such τ with $e(\tau) = e$, we will write $\tau(\sigma) = \tau$. This indicates that τ is σ's mother. σ has outcomes in decreasing order of priority ∞, d and f, so that $\sigma\widehat{\ }\infty$, $\sigma\widehat{\ }d$, $\sigma\widehat{\ }h$ and $\sigma\widehat{\ }f$ are the nodes on the tree directly extending σ. Additionally, we will insert a $2\omega^*$ sequence of outcomes of the form $\langle h, \cdot, \cdot, \cdot\rangle$ with another ω^* sequence of outcomes of the form $\langle k_1, k_2, \infty\rangle$ left of these, between d and f, in the construction below. The numbers following h will be determined by the priority assignment. Finally we will have nodes γ on the tree devoted to solving N_z for some z and we will write $z(\gamma) = z$. Nodes γ will have outcomes $\ldots, 3, 2, 1$ in decreasing order of priority. The nodes γ have length $\equiv 2 \pmod 3$. We assign some basic priority ordering to the R_e, $R_{e,x,m,n}$, and the N_z. We then build the tree inductively starting at the node λ which is devoted to R_0. This then defines the *priority tree* PT.

We begin by assigning nodes to the tree in some fair way but making sure that σ's with $e(\sigma) = e$ only occur below τ nodes with $e(\tau) = e$. The only tricky part is what we do under outcomes ∞, d and h of a σ-nodes We use the device of *lists* of e.g. Soare [10] for this.

We have partial functions e, x, m, n, z from PT to $\mathbb{N}$. We will have three *lists* L_0, L_1 and L_2 devoted to mother nodes, worker nodes and N_z-nodes, respectively. These lists are defined by induction of the length l of a node $\alpha \in PT$. For $l = 0$, let $e(\lambda) = 0$ and $L_0(\lambda) = L_1(\lambda) = L_2(\lambda) = \mathbb{N}$. Declare that 0 is *active*. For $l > 0$, let $\alpha \in PT$ be of the form $\beta\widehat{\ }a$. Adopt the first case below to pertain.

Case 1. $|\beta| \equiv 0 \pmod 3$. Let $L_0(\alpha) = L_0(\beta) - \{e(\beta)\}$. Let $L_j(\alpha) = L_j(\beta)$ for $j \neq 0$. Let $\langle e, x, m, n\rangle = \mu p(p \in L_1(\beta)$ *and* e is active). Assign $R_{e,x,m,n}$ to α, with mother $\tau \subseteq \beta$ longest such that $e(\tau) = e$. Between d and f, add the outcomes $\langle h, x, m, k\rangle$, and $\langle h, m, x, k\rangle$ as ω^* sequences, in decreasing order of k.[6] Additionally, between these outcomes and d, add an ω^* sequence of outcomes of the form $\langle k_1, k_2, \infty\rangle$.

[6] Note that in the actual construction we will discover which of $x <_A m$ or $m <_A x$ holds. At the stage we discover this, we will, for simplicity, *automatically* delete the outcomes whose hypothesis *must* be wrong. That is if we discover $m <_A x$ then we would automatically delete all the $\langle h, x, m, k\rangle$. This could have been avoided by doing things more dynamically in the construction, but we feel that the chosen method is the most easily understood.

Case 2. $|\beta| \equiv 1 \pmod 3$. Let $z(\alpha) = \mu z(z \in L_2(\alpha))$ and let $\langle e, x, m, n\rangle = \langle e, x, m, n\rangle(\beta)$.

Case 2a. $a = f$. Define $L_1(\alpha) = L_1(\beta) - \{\langle e, x, m, n\rangle\}$, and $L_j(\alpha) = L_j(\beta)$ otherwise.

Case 2b. $a = \langle h, x, m, k\rangle$. Let $L_1(\alpha) = (L_1(\beta) - \{\langle e, y, p, q\rangle : y, p, q \in \mathbb{N}\}) \cup \{\langle e, k, m', n'\rangle : m', n' \in \mathbb{N}\}$. Let $L_j(\alpha) = L_j(\beta)$ for $j \neq 1$.

Case 2c. $a = d$, $a = \infty$ or $a = \langle k_1, k_2, \infty\rangle$. If $a = \infty$ or $a = \langle k_1, k_2, \infty\rangle$, let $L_1(\alpha) = (L_1(\beta) - \{\langle e, y, p, q\rangle : y, p, q \in \mathbb{N}\}) \cup \{\langle e', y, p, q\rangle : e' > e \wedge e', y, p, q \in \mathbb{N}\}$. When $a = d$, there is more that must be done to $L_1(\alpha)$. (Essentially, we must add back the worker nodes of higher global priority which will be linked over.) Let $\tau \subset \beta$ be the longest string with $e(\tau) = e$ and $|\tau| \equiv 0 \pmod 3$. Let $L_1(\alpha) = (L_1(\tau) \cap \{\langle e', y, p, q\rangle : e' < e \wedge e', y, p, q \in \mathbb{N}\}) \cup \{\langle e', y, p, q\rangle : e' > e \wedge e', y, p, q \in \mathbb{N}\}$. Declare $e' \geq e$ as inactive. Let $L_0(\alpha) = L_0(\beta) \cup \{e' : e' > e\}$, and $L_2(\alpha) = L_2(\beta) \cup \{z : z \geq e\}$.

Case 3. $|\beta| \equiv 2 \pmod 3$. Then $\alpha = \beta\hat{\ }k$ for some $k \in \omega^*$. Let $L_2(\alpha) = L_2(\beta) - \{z(\beta)\}$, and $L_j(\alpha) = L_j(\beta)$ otherwise. Now, we will assign $e_0 = \min L_0(\alpha)$ to α, *unless* there are still $z < e_0$ in $L_2(\alpha)$ or $\langle f, y, p, q\rangle < e_0$ in $L_1(\alpha)$. In that case, we will not assign *any* R_e to α. (The point of this process is that we do not want R_e appearing on a branch of the tree *before* N_z if the global priority of R_e is *lower* than that of N_z, which might perhaps occur after we have restarted some requirement. Also, we do not want infinitely many nodes assigned to $\langle f, y, p, q\rangle$ to be linked over.) For simplicity, we will suppose that this clause is not invoked and each node of the priority tree is actually doing some job.[7] Declare the assigned e_0 to be active.

This concludes the assignment of priorities and the definition of the priority tree. Below a mother node τ devoted to solving R_e, we can define the τ-region as the collection of σ such that $\tau \subset \sigma$, and $\tau(\sigma) = \tau$. It is easy to see that if σ is in the τ-region, for some τ with $e = e(\tau)$, then there are no nodes ν with $e(\nu) \leq e$ and $\tau \subseteq \nu\hat{\ }a \subseteq \sigma$ for $a \in \{d, \infty, \langle\cdot, \cdot, \infty\rangle\}$. The following lemma is straightforward.

Lemma 17 (Finite injury along any path lemma) *For every path $h \in [PT]$ and $e, z \in \mathbb{N}$,*

(i) $(\exists^{<\infty}\alpha \subset h)(e(\alpha) = e \wedge h(|\alpha|) \in \{d, \infty, \langle\cdot, \cdot, \infty\rangle\})$,
(ii) $(\exists^{<\infty}\alpha \subset h)(|\alpha| \equiv 0 \pmod 3 \wedge e(\alpha) = e)$,
(iii) $(\exists^{<\infty}\alpha \subset h)(z(\alpha) = z \wedge |\alpha| \equiv 2 \pmod 3)$.

We can then define the *final e-mother*, and *final e-region* of a path h on PT, as the longest $\tau \subset h$ with $e(\tau) = e$, etc.

[7] Otherwise, we would need to add some clause to Case 1, saying that if $e(\beta)$ does not exist, do nothing.

The construction below will proceed in substages. We will append a subscript t to a parameter G, so that G_t denotes the value of G at substage t of the construction. As usual all parameters hold their value unless they are initialized. When initialization occurs they become undefined, or are set to zero as the case may be. We will append a parameter $[s]$, when necessary, to denote stage s. We may write (s, t) to denote substage t of stage s.

If we visit a node ν at stage (s, t) we will say that (s, t) is a *genuine* ν*-stage*. It might be that we do not visit ν at stage (s, t), rather we visit some $\widehat{\nu}$ extending ν. In this case we say that (s, t) is a ν-stage, and hence a ν-stage may not be genuine. In fact, should we put in place some permanent link (τ, σ) with $\tau \subset \nu \subset \sigma$, then ν might only ever be visited finitely often. However, this is when $\sigma\widehat{\ }d$ is the true outcome for some higher priority τ, and we would claim that a new version of ν would live below outcome d of σ. We will eventually define the genuine true path of nodes that are on the leftmost path visited infinitely often and for which there are infinitely many genuine stages. We will need to prove that each requirement has a representative on the genuine true path. We return to this point later.

We will use the following parameters within the construction:

(i) For a worker node σ, $F(\sigma, s) \in \{\infty, d, h, f\}$ is the *current state of the* σ*-module*. This state is initialized to f.
(ii) For a worker node σ, $r(\sigma, s)$ will denote a (possibly empty) interval in the ordering which it desires to preserve.
(iii) $TP[s + 1]$ will be the approximation to the *true path* at stage $s + 1$. Naturally $TP[s+1]_t$ denotes the approximation to this approximation at substage t of stage $s + 1$.

The collected restraint, at stage $s+1$, of all worker nodes which a node β must respect is $\bigcup\{r(\sigma, s + 1) : \sigma\widehat{\ }d \leq_L \beta\}$. For any point x, we define $B_x(\beta)[s + 1]$ to be the block of $\bigcup\{r(\sigma, s + 1) : \sigma\widehat{\ }d \leq_L \beta\}$ which contains x.

The Construction

Stage $s + 1$. This will proceed in substages $t \leq s$. As usual, we will generate a set of accessible nodes, $TP[s + 1]_t$, and will automatically initialize nodes α right of $TP[s + 1]_t$.

Substage 0. Define $TP[s + 1]_0 = \lambda$, the empty string.

Substage $t + 1$. We will be given a string $\beta = TP[s + 1]_t$. Adopt the first case to pertain below.

Case 1. $|\beta| \equiv 0 \pmod 3$.

Subcase 1a. There is a link (β, σ) for some node σ.

Action: Our action is to set $TP[s+1]_{t+1} = \sigma$ and go to substage $t+2$ (unless of course $t = s$, in which case go to stage $s+1$). We refer to this action as *traveling the link.*

Subcase 1b. Otherwise. Set $TP[s+1]_{t+1} = \beta\widehat{\ }o$.

Case 2. $|\beta| \equiv 1 \pmod 3$. Let $\langle e, x, m, n\rangle = \langle e, x, m, n\rangle(\beta)$. See if $\varphi_e(x, s+1) = m$ and $\varphi_e(m, s+1) = n$.

Subcase 2a. No.

Subsubcase 2a.1. No and $F(\beta, s+1)_t = \infty$, f or $\langle k_1, k_2, \infty\rangle$ (for some k_1, k_2).

Action: In this case, simply set $TP[s+1]_{t+1} = \beta\widehat{\ }f$ and go to substage $t+2$. Set $F(\beta, s+1)_{t+1} = f$

Subsubcase 2a.2. No and $F(\beta, s+1)_t = d$ (and hence we have just traveled a link $(\tau(\beta), \beta)$).

Action: Remove the link. Set $TP[s+1]_{t+1} = \beta\widehat{\ }\infty$. Set $F(\beta, s+1)_{t+1} = \infty$.

Subsubcase 2a.3. No and $F(\beta, s+1)_t = \langle h, x, m, k\rangle$ for some k. Let k_1 be the left end point and k_2 the right end point of $B_k(\beta)[s+1]_t$, the current block containing k.

Action: Set $TP[s+1]_{t+1} = \beta\widehat{\ }\langle k_1, k_2, \infty\rangle$. Set $F(\beta, s+1)_{t+1} = \langle k_1, k_2, \infty\rangle$.

For the "yes" cases below we assume, without loss of generality, that $x <_A m <_A n$.

Subcase 2b. Yes and x is not the right end point of $B_x(\beta)[s+1]_t$ (or left, in the case that $n <_A m <_A x$).

Action: Let k be the right end point of $B_x(\beta)[s+1]_t$. Set $TP[s+1]_{t+1} = \beta\widehat{\ }\langle h, x, m, k\rangle$ and $F(\beta, s+1)_{t+1} = \langle h, x, m, k\rangle$. Enumerate a "pseudo-$N$-requirement" at α saying that $B_x(\beta)[s+1]_t$ will be a complete block. Enumerate points immediately left and right of k_1 and k_2, the left and right end points of $B_k(\beta)[s+1]_t$, respectively. Go to substage $t+2$. [8]

Subcase 2c. Yes and there is an $N_{z(\nu)}$ with $\nu \subset \tau(\beta)$ (so it has higher global priority than β) such that $B_{z(\nu)}(\nu)[s+1]_t \cap [m, n] \neq \emptyset$, or there is a worker node σ with higher global priority (i.e. $\tau(\sigma) \subset \tau(\beta)$) which is not being linked over and such that either

- $\sigma\widehat{\ }\infty \subset \beta$ and $B_{x(\sigma)}(\sigma)[s+1]_t \cap [m, n] \neq \emptyset$,
- $\sigma\widehat{\ }\langle h, \cdot, \cdot, k\rangle \subset \beta$ and $B_k(\sigma)[s+1]_t \cap [m, n] \neq \emptyset$, or
- $\sigma\widehat{\ }\langle k_1, k_2, \infty\rangle \subset \beta$ and $[k_1, k_2] \cap [m, n] \neq \emptyset$.

[8] This pseudo-requirement will be invoked each time we visit β. From some point onwards, if β is on the true path, then $B_k(\beta)[s+1]_t$ comes to a limit, and this will create a finite block.

Action: Play an "h" outcome as in Subcase 2b, except this time let k be the right end point of $B_n(\beta)[s+1]_t$.

Subcase 2d. Yes, and none of the subcases above pertain (so we can play the d outcome without conflict).

Action: If there is already a link $(\tau(\beta), \beta)$ which we have just traveled, simply assign $TP[s+1]_{t+1} = \beta\hat{\ }d$, and go to substage $t+2$. If there is no such link, create a link $(\tau(\beta), \beta)$. Let $r(\beta, s+1)_{t+1} = [m, n]$. Enumerate $|[m, n]|$ many new points immediately left and right of $B_x(\beta)[s+1]_t$, and *a fortiori* between x and m (in fact, x is the right end point of $B_x(\beta)[s+1]_t$ because we are not in Subcase 2b). Set $TP[s+1]_{t+1} = \beta\hat{\ }d$, and go to substage $t+2$. Set $F(\beta, s+1)_{t+1} = d$.

Case 3. $|\beta| \equiv 2 \pmod 3$. Let $z = z(\beta)$. Let $TP[s+1]_{t+1} = \beta\hat{\ }|B_z(\beta)[s+1]_{t+1}|$. Put points immediately left and right of this block into the ordering. Go to substage $t+2$ unless $t = s$, in which case go to stage $s+2$.

End of Construction

We now verify the construction. One important observation is that once a link $(\tau(\beta), \beta)$ has been created, Subcases 2b and 2c cannot be played until the link is destroyed. This is because the restraints they test cannot change without the link being reset. Therefore, the only cases that can be invoked after traveling a link are 2a.2 and 2d.

Note that since links are created from worker nodes to mother nodes, the following can be established by induction on construction and the formation of the priority tree.

Lemma 18 *For every stage u, if (τ_1, σ_1) and (τ_2, σ_2) are links that exist at stage u, then if $\tau_1 \subset \tau_2 \subset \sigma_1$, it is the case that $\tau_2 \subset \sigma_2 \subset \sigma_1$. That is, there are no crossed links.*

Proof. Suppose not, so $\tau_1 \subset \tau_2 \subset \sigma_1 \subset \sigma_2$. First assume that (τ_1, σ_1) was created at a stage (s, t) before (τ_2, σ_2). We argue that (τ_2, σ_2) cannot be created at any stage $s' \geq s$ during which (τ_1, σ_1) still exists. This is because, if we have traveled the link (τ_1, σ_1) or just created this link, we must play the outcome $\sigma_1\hat{\ }d$ or $\sigma_1\hat{\ }\infty$ via one of Subcase 2d or Subsubcase 2a.2.

In the case that we play outcome $\sigma_1\hat{\ }d$, nodes below this one either have mothers above τ_1 (if they have higher global priority), or have mothers below $\sigma_1\hat{\ }d$. Thus τ_2 could not be in the critical region. In the case that we would play the outcome ∞, we would remove (τ_1, σ_1) (and again the nodes below the outcome ∞ either have mothers above τ_1 or below $\sigma_1\hat{\ }\infty$). In either case, (τ_1, σ_1) and (τ_2, σ_2) do not coexist.

If, on the other hand, (τ_2, σ_2) was created before (τ_1, σ_1), then each time we hit τ_2, we would travel the link down to σ_2 and skip over σ_1. Thus this case can not occur either. □

The same reasoning also gives the following lemma.

Lemma 19 *Suppose that we create a link (τ, σ) at stage s. Then there is a substage t of stage s such that $TP[s]_t = \tau$.*

Proof. If a link (τ, σ) is created at (s, t'), then $TP[s]_{t'} = \sigma$. Suppose that for no $t \leq t'$, $TP[s]_t = \tau$. Then since σ is accessible at this stage, it follows that τ has been skipped over at stage s. Therefore there is a link (α, ν) with $\alpha \subset \tau \subset \nu \subset \sigma$. This contradicts Lemma 18. □

Define the true path TP via $\lambda < TP$ and if $\gamma \leq TP$, $\gamma\widehat{\ }a \leq TP$ iff there exist infinitely many stages s where $\gamma\widehat{\ }a < TP[s]_t$, and there are only finitely many stages (s,t) where $TP[s]_t <_L \gamma\widehat{\ }a$. We then define the *genuine* true path GTP as the collection of $\sigma \leq TP$ such that $\sigma = TP[s]_t$ for infinitely many stages (s,t). These are the nodes that are actually *visited* infinitely often.

An immediate consequence of Lemma 19 is the following.

Lemma 20 *Suppose that σ is a worker node on GTP. Then $\tau(\sigma)$ is on GTP.*

It is not altogether obvious that either the TP or GTP exist, and this is the gist of the following which is proven by simultaneous induction on β.

Lemma 21 (Truth of Outcome Lemma)

(i) *If β is on GTP, then there exists a ν extending β on GPT. Hence GTP exists and is infinite. More precisely:*

(ii) *Suppose that β is on GTP, and $|\beta| \equiv 0 \pmod 3$. At any stage s, there is at most one link with mother node β. Furthermore, either $\beta\widehat{\ }o \in$ GTP, or there exists a $\sigma\widehat{\ }d \in$ GTP with $\tau(\sigma) = \beta$, and a stage s such that, for all $s' > s$, there is a link (β, σ).*

(iii) *Suppose that $\beta \in$ GTP, and $|\beta| \equiv 1 \pmod 3$. Then for some a, $\beta\widehat{\ }a \in$ GTP. For the following let $\langle e, x, m, n\rangle = \langle e, x, m, n\rangle(\beta)$.*

(iiia) *If $a = \infty$, then φ_e is not a self-embedding, and either $\lim_s \varphi_e(x,s)$ or $\lim_s \varphi_e(m,s)$ does not exist. Moreover, $|[x,m]| = \infty$. Additionally, $B_x = \lim\{B_x(\beta)[s] : s \text{ a } \beta\text{-stage}\}$ exists, and its end points are limit points.*

(iiib) *If $a = d$, then there exists some s such that for all $s' > s$, there is a link $(\tau(\beta), \beta)$. This link is never removed. φ_e is not a nontrivial self embedding. $\lim_s(r(\beta, s))$ exists and is finite (that is, is a finite interval).*

(iiic) *If $a = \langle k_1, k_2, \infty\rangle$, then φ_e is not a self-embedding, and either $\lim_s \varphi_e(x,s)$ or $\lim_s \varphi_e(m,s)$ does not exist. $\lim_s B_{k(\beta)}(\beta)[s]$ exists and is finite. k_1 is a left limit point and k_2 is a right one.*

(iiid) *If $a = \langle h, x, m, k\rangle$ (or the other version), then $\lim_s \varphi_e(x,s) = m$ and $\varphi_e(m,s) = n$. $\lim_s B_k(\beta)[s]$ exists and is finite. The left and right ends of this block are limit points, and hence k is a right limit point.*

(iiie) *If $a = f$, then only finitely often will both $\varphi_e(x,s) = m$ and $\varphi_e(m,s) = n$.*

(iv) *Suppose that $\beta \in$ GTP, and $|\beta| \equiv 2 \pmod 3$. Then $\lim_s B_{z(\beta)}(\beta)[s]$ exists and its end points are limit points. Furthermore, $\beta\widehat{\ }|\lim_s B_{z(\beta)}(\beta)[s]|$ is on GTP.*

Proof. Suppose that $\beta \in$ GTP. Also suppose that we have reached a stage s_0 after which we are never left of β, and furthermore, that for all $\rho \leq \beta$, restraints and blocks have come to their limits.

(ii) Suppose that $|\beta| \equiv 0 \pmod 3$. We know that there are infinitely many genuine β-stages, i.e. stages (s, t) where $TP(s)_t = \beta$. Now at any such stage (s, t) if there is no link with mother β, then as seen in Subcase 1b of Case 1 of the construction, we will automatically play $\beta\widehat{\ }o$, which is the only outcome available. Consequently, if there exist infinitely many stages (s, t) where there is no link then $\beta\widehat{\ }o \in$ GTP.

Now note that if we create a link (β, σ), then we will play the outcome $\sigma\widehat{\ }d$. Similarly, if we traverse the link (β, σ), then we either play the outcome $\sigma\widehat{\ }d$ or $\sigma\widehat{\ }\infty$ in the construction. (Subsubcases 2a.2 and Subcase 2d are the only cases where links have been traveled to worker nodes.) Now by the construction of the priority tree, below the outcome $a = d$ or $a = \infty$ we have removed e from $L_0(\sigma\widehat{\ }a)$, as well as $\langle e, y, p, q\rangle$ for all $y, p, q \in \mathbb{N}$ from $L_1(\sigma\widehat{\ }a)$. Hence there are *no* worker nodes σ' with $\tau(\sigma') = \beta$ below $\sigma\widehat{\ }a$. It is true that e-worker nodes can re-occur for some σ' extending $\sigma\widehat{\ }a$, but this can only happen after e is replaced on the L_0 list for some extension ν of $\sigma\widehat{\ }a$. Should this happen, before we would have any e-worker nodes, we would need a new e-mother node β' extending $\sigma\widehat{\ }a$ and then any link from an e-worker node σ' would be of the form (β', σ').

The conclusion is that at any stage s, there is at most one link with mother node β. In particular, if the link (β, σ) is erased at stage (s, t) then it cannot be replaced by another link with mother node β before the end of stage s.

Now assume that there are only finitely many stages where there is no link from β, so each time we hit β we would travel a link $(\beta, \cdot)$. Suppose that s_1 is the stage after which there is always a link at β and suppose the link at s_1 is (β, σ). We claim that this link is immortal. When we travel the link we either play the outcome $\sigma\widehat{\ }d$ or $\sigma\widehat{\ }\infty$. If the hypotheses have changed when we hit σ so that either $\varphi_e(x(\sigma), s)_{t+1} \neq m(\sigma)$ or $\varphi_e(m(\sigma), s)_{t+1} \neq n(\sigma)$, then we would invoke Subsubcase 2a.2, and hence play the outcome $\sigma\widehat{\ }\infty$. When we play that at stage $(s, t+1)$, we would remove the link and, by the argument above, there would not be a link from β the next time it is visited, contradicting our assumption. Therefore, we know that $\sigma\widehat{\ }d$ is always played after stage s_1 and the link (β, σ) is preserved.

We conclude that if β has length $\equiv 0 \pmod 3$ and $\beta\widehat{\ }o \notin$ GTP, then there is some permanent link (β, σ) and $\sigma\widehat{\ }d \in$ GTP. This proves (ii).

To prove (iii), suppose that $|\beta| \equiv 1 \pmod 3$, and that $\beta \in$ GTP. The construction ensures that each time we have $TP(s)_t = \beta$, we will play *some* outcome of β. However, since there are infinitely many outcomes, we need to prove that the lim inf of the outcomes exists.

Clearly, after s_0, if, whenever we hit β it is the case that $\varphi_e(x,s) \neq m$ or $\varphi_e(m,s) \neq n$ then we will always play f. (From this (iiie) follows, also.) Hence, the only time we ever play something *other* than f is if the hypothesis of β looks correct. Suppose that $\beta^\frown f$ is not in GTP. Then we need some β-stage $s > s_0$ where $\varphi_e(x,s) = m$ and $\varphi_e(m,s) = n$. Since β is on GTP, it follows by Lemma 20 that $\tau(\beta) \in$ GTP.

Now since β's hypothesis appears correct, we would invoke one of options 2b-2d. We consider these case by case.

Case 1a. 2b is invoked. Then at stage (s,t), the interval $[x,m]$ is covered by the union of the restraints $r(\sigma_1,s)_t$ for all $\sigma_1{}^\frown d \leq_L \beta$. At stage s, we would play the outcome $\beta^\frown\langle h,x,m,k\rangle$ where k is determined by $\rho \leq_L \beta$. Notice that, by induction and choice of s_0, these restraints, and hence k, have come to a limit. Therefore, each time 2b is invoked we would play the same outcome $\beta^\frown\langle h,x,m,k\rangle$. If 2b is invoked cofinitely, then $\beta^\frown\langle h,x,m,k\rangle \in$ GTP.

Now suppose that 2b is not invoked cofinitely many times. Then infinitely often when we hit β it must be the case that $\varphi_e(x,s) \neq m$ or $\varphi_e(m,s) \neq n$. Then at such a stage we would play $\langle k_1,k_2,\infty\rangle$, again for a fixed k_1,k_2. Thus in that case $\sigma^\frown\langle k_1,k_2,\infty\rangle \in$ GTP.

Finally, note the truth of the outcomes in either case. In particular, if 2b is invoked cofinitely many times, then from some point on whenever we access β we play $\beta^\frown\langle h,x,m,k\rangle$ which means that the hypothesis is proven correct. That is, $\varphi_e(x) = m$ and $\varphi_e(m) = n$, and as we have already observed, $\lim_s B_k(\beta)[s]$ exists.

If $F(\beta,s) \neq \langle h,x,m,k\rangle$ infinitely often, then we have seen that $\langle k_1,k_2,\infty\rangle$ is played infinitely often.

(It is important to note that after stage s_0, once we have invoked 2b, our only possible options at β stages are to play $\beta^\frown\langle h,x,m,k\rangle$, $\beta^\frown f$ or $\beta^\frown\langle k_1,k_2,\infty\rangle$.)

Case 1b. 2b is never invoked, but 2c is invoked. The reasoning is the same as in Case 1a above.

Because $\beta^\frown\langle h,x,m,k\rangle$ and $\langle k_1,k_2,\infty\rangle$ are only played after 2b or 2c is invoked, we have proved (iiic) and (iiid).

Case 2. 2b and 2c are never invoked. Hence, 2b.4 is invoked. In this case, we will create a link $(\tau(\beta),\beta)$, and the outcome will be $\beta^\frown d$.

Again either this link exists cofinitely often, or it is removed infinitely often.

In the former case, $\beta^\frown d \in$ GTP. Furthermore we see that our action at the last stage we create the link we ensure that $|x,m| > |[m,n]|$, and by construction this will be preserved. Thus φ_e cannot be a self-embedding. Hence (iiib) holds.

In the latter case, $\beta^\frown\infty \in$ GTP, and (iiia) clearly holds, since infinitely often we will increase the cardinality of $[x, m]$. Additionally we do so by making the end points of $B_x(\beta)[s]$ limit points, $B_x(\beta)[s]$ coming to a limit by the inductive hypothesis at β-stages.

Finally, if $|\beta| \equiv 2 \pmod 3$, then at stage s_0, we will have determined $B_{z(\beta)}(\beta)$. Whenever we visit β we will add points left and right of this block. Therefore $\beta^\frown|B_{z(\beta)}(\beta)| \in$ GTP. $\square$

The final lemma we need is that all requirements are actually met.

Lemma 22 (Golden Path Lemma)

(i) For all e, there is a final e-mother $\tau \in$ GTP.
(ii) There are at most finitely many e-worker nodes σ in the final e-region below τ with $\sigma^\frown\langle h, \cdot, \cdot, \cdot\rangle \in$ GTP.
(iii) Suppose that there is some worker node $\sigma \in$ GTP with $\tau(\sigma) = \tau$ such that $\sigma^\frown f$ is not on GTP. Then φ_e is not a self-embedding.
(iv) If φ_e is a self-embedding, then it must be trivial. Hence R_e is met.
(v) For all z, there is a node ν such that $|\nu| \equiv 2 \pmod 3$, $z = z(\nu)$ and $\nu \in$ GTP. Hence N_z is met.

Proof. By the Finite injury along any path Lemma, we know that there is definitely a final e-mother $\tau <$ GTP. The question is why should it be on GTP. But this follows by the same reasoning as Lemma 20.

Now note that if there is a worker node σ extending τ with $\tau(\sigma) = \tau$ and $\sigma^\frown d$, $\sigma^\frown\infty$ or $\sigma^\frown\langle h_1, h_2, \infty\rangle$ on GTP, then no further e-worker nodes occur on TP. Furthermore, by parts (iiib), (iiia) and (iiic) of Lemma 21, R_e is satisfied. So for (ii)–(iv) we can restrict ourselves to the cases where all e-worker nodes below τ on GTP must have f or $\langle h, \cdot, \cdot, \cdot\rangle$ outcomes on GTP.

To prove (ii), we examine the two distinct cases in the construction in which $\langle h, \cdot, \cdot, \cdot\rangle$ outcomes are played: Subcases 2b and 2c. Let σ be a worker node extending τ with $\tau(\sigma) = \tau$, $\langle e, x, m, n\rangle = \langle e, x, m, n\rangle(\sigma)$ and $\sigma^\frown\langle h, x, m, k\rangle \in$ GTP. One reason that we might play $\sigma^\frown\langle h, x, m, k\rangle$ is if x is not the right end point of $B_x(\beta)[s+1]_t$. This is Subcase 2b of the construction.

We want to prove that Subcase 2b can only account for, at most, the first $\langle h, \cdot, \cdot, \cdot\rangle$ outcome on GTP of any e-worker extending τ. Assume that there were another e-worker node σ' extending τ and $\sigma'^\frown\langle h, x', m', k'\rangle \in$ GTP. Furthermore, assume that σ' is the least such predecessor of σ on GTP. Therefore, $x = k'$ and no requirement can preserve a block containing x (because all higher priority requirement are done acting). Hence x *is* the right end point of $B_x(\sigma)[s+1]_t$ and Subcase 2b is not invoked.

This leaves Subcase 2c, the only other way that $\sigma^\frown\langle h, x, m, k\rangle$ can be played. This is invoked if either there is an $N_{z(\nu)}$ with $\nu \subset \tau(\beta)$ such that

- ◦ $B_{z(\nu)}(\nu)[s+1]_t \cap [m,n] \neq \emptyset$,

or there is a worker node σ' with $\tau(\sigma') \subset \tau(\sigma)$ which is not being linked over, such that either

- ◦ $\sigma'^\frown\infty \subset \sigma$ and $B_{x(\sigma')}(\sigma')[s+1]_t \cap [m,n] \neq \emptyset$,
- ◦ $\sigma'^\frown\langle h,\cdot,\cdot,k\rangle \subset \sigma$ and $B_k(\sigma')[s+1]_t \cap [m,n] \neq \emptyset$, or
- ◦ $\sigma'^\frown\langle k_1,k_2,\infty\rangle \subset \sigma$ and $[k_1,k_2] \cap [m,n] \neq \emptyset$.

The number of such ν is clearly finite since they must be above $\tau(\sigma)$. In the second case, note that $\sigma'^\frown a \in$ GTP, where $a \in \{\infty, \langle\cdot,\cdot,\infty\rangle, \langle h,\cdot,\cdot,\cdot\rangle\}$. (This is because we excluded nodes which are being linked over.) But by part (i) of the Finite injury along any path Lemma, and by induction on the current claim in the case that $a = \langle h,\cdot,\cdot,\cdot\rangle$, for each $e' < e$ there are only finitely many such σ' with $e' = e(\sigma')$. The blocks being protected by these ν and σ' nodes stabilize at some finite stage. Let D_e be the union of these blocks, which is finite.

Once D_e has stabilized, each new $\sigma'^\frown\langle h,x',m',k'\rangle \in$ GTP with $\tau(\sigma') = \tau$ must move k' to the right past some offending interval in D_e. Because this can only happen finitely often, we have proved (ii).

(iii) As mentioned above, we can restrict ourselves to the cases where the e-worker nodes on GTP which extend τ have either f or $\langle h,\cdot,\cdot,\cdot\rangle$ outcomes on GTP. By assumption, some non-f outcome must occur. By part (ii), $\langle h,\cdot,\cdot,\cdot\rangle$ outcomes occur only finitely often, so there is a last e-worker σ extending τ such that $\sigma^\frown\langle h,x,m,k\rangle \in$ GTP, for some $x,m,k \in \mathbb{N}$.

Assume that φ_e is a self-embedding. Then it must move k. This is because, by Lemma 21, $\lim_s \varphi_e(x,s) = m$ and $\lim_s \varphi_e(m,s) = n$ and (without loss of generality) $x <_A m <_A n$, so both x and n are moved to the right. But k is the right end point of the finite block $\lim_s B_k(\beta)[s]$, which by construction, contains either x or n (depending on whether Subcase 2b or Subcase 2c is being invoked). Therefore, φ_e moves k. By the construction of the priority tree, there must be an e-worker node σ' for $R_{e,k,\varphi_e(k),\varphi_e(\varphi_e(k))}$ on GTP. But the hypothesis of σ' is correct, so $\sigma'^\frown f$ is not on GTP, which is a contradiction.

(iv) Assume that φ_e is a self-embedding. For distinct numbers $x,m,n \in \mathbb{N}$, there is a node devoted to $R_{e,x,m,n}$ on TP. Every time such a node is linked over, a new one is generated. By the construction of the priority tree this can only happen finitely often. Hence, there is a node σ devoted to $R_{e,x,m,n}$ on GTP. By assumption, $\sigma^\frown f \in$ GTP. Hence, for any distinct numbers $x,m,n \in \mathbb{N}$, either $\varphi_e(x) \neq m$ or $\varphi_e(m) \neq n$. Therefore, φ_e is trivial.

(v) This follows by Lemma 21, and the fact that we always have a new version of N_z below any global win for a τ of higher priority. Only finitely many of these can ever be linked over. □

This concludes the proof of the theorem.

4 Appendix: The proof-theoretical strength of the Dushnik–Miller Theorem

In this section we include a clarification of the proof of Downey and Lempp [3] which shows that the Dushnik–Miller Theorem is equivalent to ACA_0 over the base theory RCA_0. One direction is effectivizing the Dushnik–Miller proof in ACA_0. This is straightforward.

The somewhat unclear direction is their proof that "RCA_0+Dushnik–Miller" implies ACA_0. We clarify this here.

Fix any $A \subseteq \mathbb{N}$ in the given second order model for "RCA_0+Dushnik–Miller". We must show that A' exists. Let $c(x) = \mu s \geq x(A' \upharpoonright (x+1) = A'_s \upharpoonright (x+1))$ be the associated *computation function* of A'. Then Downey and Lempp construct a linear ordering in the model such that

Claim 1. *Any nontrivial self-embedding i of L can compute c.*

Let M be the universe of the given model and let $<$ denote the usual ordering. Downey and Lempp define the linear ordering L of order type $(M, <)$ in stages. They start by letting L_0 be the ordering $0 \leq_L 2 \leq_L 4 \leq_L \dots$ of all even integers in M. They establish Claim 1 by ensuring the existence of a function e (given the assumption that a nontrivial self-embedding i exists) satisfying the following

Claim 2. *There is a linear order L (which exists in the model M) for which there is a function e (not necessarily in the model M yet) such that e is strictly increasing (which is easily ensured by fiat), $e : M \to L_0$ and for all $x \in M$,*

(i) $\forall n_0, n_1 < c(x)\,(e(x) <_L n_0 <_L n_1 \to d(e(x), n_0) > d(n_0, n_1))$, *and*
(ii) $e(x+1) = \mu y \in L_0(\forall n\,(n < c(x) \to n <_L y))$,

where $d(n_0, n_1)$ is the (M-finite) distance between n_0 and n_1 in L.

Sketch of Claim 2. The existence of such an L in M is easy since L is computable in A and constructed by a finite injury priority argument, using only Σ^0_1-induction, which holds in RCA_0.

We have to maintain (i) and (ii) of Claim 2 at any stage s for all $x \leq s$ (evaluating $c(x)$ for these x's at stage s). Note that the definition of the function e is fixed by (ii) at any stage s (assuming $e(0) = 0$). The only problem arises if some number x enters A' at a stage $s > 0$, thus making (i) false. In that case, add all currently unused elements $x \leq s$ in $M - L_{s-1}$ into L_s just to the left of $e(x)$, and add sufficiently many unused elements $x > s$ in $M - L_{s-1}$ into L_s just to the right of $e(x)$ to make (i) true. Note that this action will not interfere with keeping (i) satisfied for any $x' < x$. □

Sketch of Claim 1. We argue that the existence of i for L in M proves the existence of e and c in M, which will finish the proof since A' can be computed from A and c. The first step is now that i is strictly monotonic, and so by Σ^0_1-induction again, we have that $x <_L i(x)$ from some x_0 on. By the same induction, we have that any L-initial segment is M-finite (i.e., is M-bounded). Now nonuniformly fix x_0 such that $e(x_0) <_L ie(x_0)$. Arguing model-theoretically, one can deduce from (i) of Claim 2 that $c(x) \leq \max(ie(x), i^2e(x))$, so from $e(x)$ and i we can compute $c(x)$. But then from (ii) of Claim 2, we see that we can compute $e(x+1)$ from $c(x)$. The latter two facts are most easily established model-theoretically first, and then we argue that the reductions which compute $c(x)$ from $e(x)$ and i, and $e(x+1)$ from $c(x)$, are actually quite trivial, so we have e and c in the model M. □

The reader should note that the Downey–Lempp proof actually ensures that RCA_0 proves that the Dushnik–Miller Theorem for orderings of type ω is equivalent to ACA_0.

References

[1] R. Downey *Computability Theory and Linear Orderings*, in *Handbook of Recursive Mathematics* (ed. Ershov, Goncharov, Nerode and Remmel), Vol 2, North Holland, (1998), 823–977.

[2] R. Downey, G. LaForte, and R. Shore, *Decomposition and infima in the c.e. degrees*, *Journal of Symbolic Logic*, Vol. 68 (2003), 551–579.

[3] R. Downey and S. Lempp, *On the proof theoretical strength of the Dushnik–Miller theorem for countable linear orderings*, in, *Recursion Theory and Complexity* (ed. Arslanov and Lempp), de Gruyter, 1999, 55–58.

[4] R. Downey and M. Stob, *Minimal pairs in initial segments of the recursively enumerable degrees*, *Israel Journal of Mathematics*, Vol. 100 (1997), 7–27.

[5] B. Dushnik and E. Miller, *Concerning similarity transformations of linearly ordered sets*, *Bull. Amer. Math. Soc.*, Vol. 46 (1940), 322–326.

[6] C. Jockusch and R. Soare, Π^0_1 *classes and degrees of theories*, *Trans. Amer. Math. Soc.*, Vol. 173 (1972), 33–56.

[7] S. Lempp, A. Morozov, C. McCoy, and R. Solomon, *On self-embeddings of computable linear orderings*, in *Computability and Models* (ed. Cooper and Goncharov), Kluwer Academic/Plenum Publishers, New York, 2003, 259–265.

[8] J. Rosenstein, *Linear Orderings*, Academic Press, New York (1982).

[9] S. Simpson, *Degrees of Unsolvability: a survey of results*, pages 631–652 in *Handbook of Mathematical Logic*, ed. by J. Barwise, North Holland, Amsterdam (1977).

[10] R. Soare, *Recursively Enumerable Sets and Degrees*, Springer-Verlag, New York (1987).