The Δ^0_2-Spectrum of a Linear Order

Russell Miller*

September 1, 2004

Abstract

Slaman and Wehner have constructed structures which distinguish the computable Turing degree $\mathbf{0}$ from the noncomputable degrees, in the sense that the spectrum of each structure consists precisely of the noncomputable degrees. Downey has asked if this can be done for an ordinary type of structure such as a linear order. We show that there exists a linear order whose spectrum includes every noncomputable Δ^0_2 degree, but not $\mathbf{0}$. Since our argument requires the technique of permitting below a Δ^0_2 set, we include a detailed explanation of the mechanics and intuition behind this type of permitting.

1 Introduction

Definition 1.1 The *spectrum* $\text{Spec}(\mathcal{A})$ of a structure $\mathcal{A}$ is the class of Turing degrees of presentations of $\mathcal{A}$,

$$\text{Spec}(\mathcal{A}) = \{\deg(\mathcal{B}) : \mathcal{B} \cong \mathcal{A}\}.$$

(Here the degree of a structure $\mathcal{B}$ is the supremum of the degree of its universe and the degree of its open diagram. For our purposes, the universe will generally be ω.)

*This article is the first chapter of a Ph.D. thesis at the University of Chicago under the supervision of Robert I. Soare. It was published in *The Journal of Symbolic Logic* **66** (2001), pp. 470-486. A summary of these results was presented at the AMS conference in Gainesville, FL, in March 1999. Many thanks go to Soare, and also to Carl G. Jockusch, Jr. for useful conversations.

Slaman [14] and Wehner [17] have recently each constructed a countable first-order structure $\mathcal{A}$ such that $\text{Spec}(\mathcal{A}) = \mathbf{D} - \{\mathbf{0}\}$, where $\mathbf{D}$ is the class of all Turing degrees and $\mathbf{0}$ is the degree of the computable sets. This answers a question in [3] from Lempp, who had asked whether it was possible to distinguish the noncomputable degrees from the degree $\mathbf{0}$ in such a way. Slaman remarks that the open diagram of each of these models contains information which is common to all noncomputable real numbers, yet which is not itself computable. (In contrast, a single subset of ω with no algebraic structure cannot contain such information; the existence of a minimal pair of Turing degrees ensures that any set which is computable in every noncomputable real must itself be computable.)

The structures constructed by Slaman and Wehner were built specifically for this purpose and are not readily recognizable to most mathematicians. Downey [3] has asked whether one could do the same for better-known types of mathematical objects, particularly for linear orders. Indeed, he posed a series of questions:

Question 1.2 (Downey) *Is there a linear order whose spectrum contains every computably enumerable Turing degree except* $\mathbf{0}$*?*

Question 1.3 (Downey) *Is there a linear order whose spectrum contains every* Δ^0_2 *degree except* $\mathbf{0}$*?*

Question 1.4 (Downey) *Is there a linear order whose spectrum contains every degree except* $\mathbf{0}$*?*

We can rephrase these questions using the following terminology.

Definition 1.5 If $\mathbf{C}$ is a class of Turing degrees, the $\mathbf{C}$-*spectrum* of $\mathcal{A}$, written $\text{Spec}^{\mathbf{C}}(\mathcal{A})$, is the intersection of $\mathbf{C}$ with $\text{Spec}(\mathcal{A})$.

We will consider Σ^0_1 and Δ^0_2 as classes of degrees, not classes of sets. Thus, Question 1.2 asks whether the Σ^0_1-spectrum of a linear order $\mathcal{A}$ can be precisely the noncomputable Σ^0_1 degrees, and Questions 1.3 and 1.4 are the corresponding questions for $\text{Spec}^{\Delta^0_2}(\mathcal{A})$ and $\text{Spec}(\mathcal{A})$.

For certain common mathematical structures, the answers to such questions are negative. For instance, Downey and Jockusch have shown in [5] that any Boolean algebra $\mathcal{B}$ of low degree is isomorphic to a computable Boolean algebra,

$$\text{Spec}(\mathcal{B}) \cap \mathbf{L_1} \neq \emptyset \implies \mathbf{0} \in \text{Spec}(\mathcal{B}).$$

Hence the Σ_1^0-spectrum of a Boolean algebra cannot contain every noncomputable computably enumerable (c.e.) degree without also containing $\mathbf{0}$. (This result was extended to the low$_2$ degrees by Thurber [16] and then as far as the low$_4$ degrees by Knight and Stob [10], who proved that any Boolean algebra of low$_4$ degree is isomorphic to a computable Boolean algebra.)

However, it is known that for every noncomputable Turing degree, there exists a linear order of that degree which is not isomorphic to any computable linear order. Jockusch and Soare [8] proved this statement for noncomputable c.e. degrees, by creating a linear order which could be "separated" into countably many components, which are used to diagonalize against all possible computable linear orders. Later, Downey and Seetapun (both unpublished) independently extended this result to the noncomputable Δ_2^0 degrees. Finally, Knight proved the result for an arbitrary noncomputable Turing degree (see [3], p. 179), suggesting that a positive answer to Downey's most general question might be possible.

The argument by Jockusch and Soare is uniform in the given noncomputable c.e. set C in whose degree we wish to build a linear order with no computable copy. It does give different results, namely non-isomorphic linear orders, for different sets C. The same is true of Downey and Seetapun's results, which use the same basic module. Therefore these results do not answer any of Downey's questions.

In this paper we modify the Jockusch-Soare basic module so that for any two noncomputable c.e. sets C and D, it produces isomorphic copies of the same linear order. Also, we modify and develop the method of Δ_2^0-permitting so that the basic module can handle any noncomputable Δ_2^0 set C, while still producing isomorphic linear orders regardless of the choice of C. We use this new basic module in Section 4 to prove:

Theorem 4.1 *There exists a linear order $\mathcal{A}$ which has a copy in every noncomputable Δ_2^0 degree, but no computable copy,*

$$Spec^{\Delta_2^0}(\mathcal{A}) = \Delta_2^0 - \{\mathbf{0}\}.$$

Furthermore, this order may be taken to be of the form

$$\mathcal{A} = \sum_{i \in \omega} (\mathcal{S}_i + \mathcal{A}_i),$$

where each $\mathcal{S}_i \cong 1 + \nu + i + \nu + 1$ and each $\mathcal{A}_i$ is either ω or of the form $c_i + \omega^* + \omega$ for some $c_i \in \omega$.

(Here ν represents the countable dense linear order with end points.)

This answers Downey's Questions 1.2 and 1.3. Question 1.4 is still open, and is discussed in the final section.

Although the method of Δ^0_2-*permitting* has been occasionally used in computability theory, the literature on it is far less complete than that on Σ^0_1-permitting. Perhaps the most useful reference for Δ^0_2-permitting has been the twenty-year-old paper of Posner [11]. Therefore, we devote Section 2 to a revision, updating, and expansion of Posner's presentation. This includes an explanation of the intuition behind the method, with examples, and a general lemma, omitted from Posner's paper, explaining why one must receive permission infinitely often.

The rest of the paper serves the dual purpose of answering Downey's question and providing a full example of Δ^0_2-permitting. In Section 3 we give the basic module for the construction, with Δ^0_2-permitting prominently used and explained, and in Section 4 we present the complete construction.

We use the notation of Soare [15] regarding Turing degrees and computability, and that of Rosenstein [13] for linear orders. (Thus ω^* represents the reverse order of ω, i.e. the order type of the negative integers.) When $\{C_s : s \in \omega\}$ is a computable approximation for a set C, we will usually just write $\{C_s\}$ to stand for the entire approximation. Also, we use the symbol $S \upharpoonright\!\upharpoonright x$ to denote $S \upharpoonright (x+1)$, the restriction of the subset $S \subseteq \omega$ (viewed as a function) to the elements $0, 1, \ldots x$.

2 Δ^0_2 Permitting

Δ^0_2 permitting is not as transparent as c.e. permitting. Posner [11] has succinctly outlined the differences, as well as the tree approach we use to overcome them. In the c.e. case, we can be sure at least that every element that has entered the permitting set C will stay there; for a Δ^0_2 set C, there is no such guarantee for any element. Let $\{C_s\}_{s\in\omega}$ be a computable approximation of the permitting set, and suppose A is the C-computable set we wish to build. The permitting condition is actually the same for both the c.e. case and the Δ^0_2 case, and suffices to ensure that $A \leq_T C$:

Requirement 2.1 (Permitting Condition) *If* $C_s\restriction m = C_t\restriction m$ *and* $m \leq \min(s,t)$*, then* $A_s\restriction m = A_t\restriction m$*.*

However, for a c.e. permitting set C, we know that permission, once given, will never be withdrawn. That is, if $C_s\restriction m \neq C_{s+1}\restriction m$, then we must also have $C_s\restriction m \neq C_t\restriction m$ for every $t > s$, and therefore we never again have to worry about making $A_t\restriction m$ equal to $A_s\restriction m$. In the Δ^0_2 case, on the other hand, it is perfectly possible to have $C_s\restriction m \neq C_{s+1}\restriction m$ and $C_s\restriction m = C_t\restriction m$ for some $t > s+1$. If so, we must undo everything we have done to $A\restriction m$ since stage s and ensure that $A_t\restriction m = A_s\restriction m$.

The easiest way to visualize our solution to this difficulty is by use of a tree, called the approximation-tree for C, which we define below after setting up some machinery. For $s > 0$, let:

$$x_s = \max\{x : (\exists t < s)[x \leq t \ \&\ C_s\restriction x = C_t\restriction x]\},$$

$$t_s = \min\{t : x_s \leq t < s \ \&\ C_s\restriction x_s = C_t\restriction x_s\}.$$

Thus x_s is the greatest length of agreement of C_s with any preceding stage, and t_s is that preceding stage (or the first such stage, if there is more than one). Notice that we always have $x_s \leq t_s$. (The requirement $x \leq t$ in the definition of x_s averts the possibility of x_s being infinite, if there should be a stage $t < s$ such that $C_t = C_s$.)

The *approximation-tree* $T(\{C_s\})$ for C is a computable tree with an integer at each node. The top node of this tree is 0, and each integer s is added to the tree as an immediate successor of t_s. The precise definition of the approximation-tree is as follows.

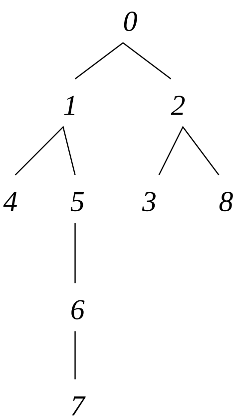
$$T(\{C_s\}) = \{\sigma \in \omega^{<\omega} : \sigma(0) = 0 \ \&\ (\forall n < (lh(\sigma) - 1))\ [\sigma(n) = t_{\sigma(n+1)}]\}.$$

(Clearly this depends on the choice of approximation $\{C_s\}$, not just on C.)

For instance, suppose that the approximations up to stage 8 are given as follows. (It is convenient to write a dash in place of the 0 or 1 for each $C_s(y)$ with $y \geq s$, since this makes it clear when the requirement $x_s \leq t_s$ comes into play. If desired, we could easily ensure that $C_s(y) = 0$ for all $y \geq s$ and still have $\{C_s\}$ be a computable approximation of C.)

s	$C_s(0)$	$C_s(1)$	$C_s(2)$	$C_s(3)$	$C_s(4)$	$C_s(5)$	$C_s(6)$	$C_s(7)$	t_s	x_s
0	–	–	–	–	–	–	–	–		
1	1	–	–	–	–	–	–	–	0	0
2	0	1	–	–	–	–	–	–	0	0
3	0	1	1	–	–	–	–	–	2	2
4	1	1	1	0	–	–	–	–	1	1
5	1	0	1	0	0	–	–	–	1	1
6	1	0	1	1	0	0	–	–	5	3
7	1	0	1	1	1	0	0	–	6	4
8	0	0	1	1	1	0	0	0	2	1

We draw the corresponding approximation-tree (restricted to the stages given above):



Lemma 2.2 *If the node t precedes the node s on the approximation-tree, then $x_t < x_s$ and $C_t \restriction x_t = C_s \restriction x_t$.*

Proof. We induct on the number of levels between s and t. If t immediately precedes s, then $t = t_s$, so $C_t \restriction x_s = C_s \restriction x_s$. Now we must have $x_t < x_s$, since otherwise C_s would agree up to x_s with a stage preceding t, contradicting the definition of t_s. Hence $C_t \upharpoonright x_t = C_s \upharpoonright x_t$.

For the inductive step, we simply note that $C_s \upharpoonright x_s = C_{t_s} \upharpoonright x_s$ and apply the inductive hypothesis to t_s. (Once again we have $x_s > x_{t_s} > x_t$.) ■

We now introduce the notion of a *true stage* for the approximation $\{C_s\}$. A true stage for this approximation is a stage s such that the length of agreement of C_s with C is greater than the corresponding length of agreement for every preceding stage,

$$(\exists x \leq s) \ [C_s \upharpoonright x = C \upharpoonright x \ \& \ (\forall t) \ [x \leq t < s \ \Rightarrow \ C_t \upharpoonright x \neq C \upharpoonright x \ ]] .$$

(For our purposes, the "length of agreement" is bounded by the stage number. Thus, we need not worry about stages t with $t < x$.)

For c.e. sets, the true stages are precisely the *nondeficiency stages*, as defined by Dekker [2], namely those such that an element a enters the set at that stage and no element less than a ever enters at any subsequent stage.

Clearly, if s is a true stage, then t_s is precisely the previous true stage. The true stages form an infinite path through the tree, indeed the only infinite path. If this path were computable, then we could compute C. (Notice, however, that the tree need not be computably bounded, so one cannot automatically compute the unique infinite path.)

Ultimately, we only need to know A_s for the true stages s. After all, there are infinitely many true stages, and the Permitting Condition (and the convergence of $\lim_s C_s$) forces $\lim_s A_s$ to converge, so any infinite increasing subsequence $\{A_{s_i} : i \in \omega\}$ of approximations must converge to A as well. Moreover, if s is a true stage, we know that $A_s \upharpoonright x_s = A \upharpoonright x_s$.

The difficulty, of course, is that it is impossible to compute the sequence of true stages, given that C is noncomputable. Our general strategy for Δ^0_2-permitting is to assume at each stage s that the node s lies on the unique infinite path through the tree, i.e. that s is a true stage. We ensure that $A_s \upharpoonright x_s = A_{t_s} \upharpoonright x_s$, thereby satisfying the Permitting Condition for s and all stages preceding it. If it turns out that s is not a true stage, then at some subsequent true stage we will have the opportunity to undo the injury done at stage s to the preceding true stages.

For a c.e. permitting set C, one characteristically uses the noncomputability of C to prove that there will be infinitely many stages at which C "gives permission" to make a change to A. The analogous result for a Δ^0_2 set C is as follows.

Lemma 2.3 **(Δ_2^0 Permission)** *Let $s_0 = 0, s_1, s_2, \ldots$ be the true stages of a computable approximation $\{C_s\}_{s\in\omega}$ of C, with $s_i < s_{i+1}$ for all i. Let $\{n_s\}_{s\in\omega}$ be a non-decreasing unbounded computable sequence. If $\{q : n_{(s_q)} > x_{(s_q)}\}$ is finite, then C is computable.*

(Notice that we conclude that permission is given at infinitely many true stages, not merely at infinitely many stages. Again, the true stages are the stages which we care about for purposes of computing A from a C-oracle.)

Proof. Suppose that there were a number k' such that for all true stages $s_q \geq k'$, we have $x_{s_q} \geq n_{s_q}$. Since $\lim_s n_s = \infty$, we can compute for each stage s the least stage t such that $n_t > s$. Define $g(s)$ to be this stage t, so the function g is computable and total and $n_{g(s)} > s$ for every s.

Let $s_q \geq k'$ be a true stage. Then $s_q = t_{s_{(q+1)}} \geq x_{s_{(q+1)}} \geq n_{s_{(q+1)}}$ (since $s_{q+1} \geq k'$). But $n_{g(s_q)} > s_q$ by definition of g, so $n_{g(s_q)} > n_{s_{(q+1)}}$. Since $\{n_s\}$ is a nondecreasing sequence, we see that $g(s_q) > s_{q+1}$. This holds as long as $s_q \geq k'$, but in fact we could redefine g at the finitely many true stages below k', to yield the following:

Sublemma 2.4 *Under the hypotheses of Lemma 2.3, there exists a computable function g such that for every true stage s_q we have $s_{q+1} < g(s_q)$.* ∎

We remark that this function g does not provide a computable bound on the approximation-tree $T(\{C_s\})$. It is possible that there is a stage s with an immediate successor t such that $t > g(s)$. Sublemma 2.4 simply asserts that in this case t cannot be a true stage.

However, this information suffices for us to compute the path of true stages in $T(\{C_s\})$. 0 is always a true stage, of course, and knowing the true stage s_q, we find all immediate successors of s_q which are less than $g(s_q)$. Say that these are $t_0, t_1, \ldots t_p$. One of these must be the next true stage s_{q+1}, and all the others have only finitely many nodes below them (by Konig's Lemma). To determine which one is the next true stage, we simultaneously find all successors of each t_j which are less than $g(t_j)$, and eliminate each t_j which has no such immediate successors. Then we find all immediate successors of those immediate successors, within the bounds provided by g, and eliminate those which have no immediate successors within the bounds. Continuing in this manner, we will eventually eliminate every t_j with only finitely many successors, and once we have only one remaining t_j, we will know that that t_j is the next true stage s_{q+1}.

(Equivalently, let

$$T' = \{\sigma \in T : (\forall n < (lh(\sigma) - 1))\ [\sigma(n+1) < g(\sigma(n))\ ]\ \}.$$

Then T' is a computable subtree of T and contains the path of true stages. But since T' is computably bounded by g, its unique infinite path must be computable.)

Thus the path of true stages is computable, and we use this to show that C is computable. Notice that on the path of true stages, we always have $x_{s_{(q+1)}} > x_{s_q}$, and thus $x_{s_q} \geq q$. Also, for all $p > q$ we have $C_{s_p} \upharpoonright x_{s_q} = C_{s_q} \upharpoonright x_{s_q}$. To compute whether $c \in C$, therefore, we need only compute the $(c+1)$-st true stage s_{c+1} and evaluate $C_{s_{(c+1)}}(c)$, since $c < x_{s_{(c+1)}}$. ∎

3 Basic Module for the Construction

Choose an arbitrary noncomputable Δ_2^0 set C with computable approximation $C = \lim_s C_s$. We give the basic module for constructing a linear order $\mathcal{A} = (A, <_{\mathcal{A}})$ of degree $\leq_T C$ which is not isomorphic to the linear order $\mathcal{B}_i$ (if any) computed by the i-th partial computable function φ_i. To achieve this, we choose an element $\hat{b}$ of the universe of $\mathcal{B}_i$ and ensure that no element of A has the same number of predecessors under $<_{\mathcal{A}}$ that $\hat{b}$ does in $\mathcal{B}_i$. This is the same result achieved by the Jockusch-Soare basic module in [8], except that the result of our construction is independent of C.

Proposition 3.1 *The basic module described below yields the following outcomes, regardless of the choice of the noncomputable Δ_2^0 set C or the computable approximation to C.*

1. *If $\hat{b}$ has exactly c predecessors in $\mathcal{B}_i$ (or more accurately, if there are exactly c elements x such that $\varphi_i(\langle x, \hat{b}\rangle)\downarrow= 1$), then the basic module constructs a linear order $\mathcal{A}$ of type $c + \omega^*$.*

2. *If $\hat{b}$ has infinitely many predecessors in $\mathcal{B}_i$, then the basic module constructs a linear order $\mathcal{A}$ of type ω.*

(Notice that each outcome ensures that $\mathcal{A} \not\cong \mathcal{B}_i$, since no element of $c + \omega^*$ has exactly c predecessors and no element of ω has infinitely many predecessors.)

The universe A of this order will be $\bigcup_s A_s$, with each $A_s = \{a_0, a_1, \dots a_s\}$. In fact we could just take $a_i = i$ for all i, but this way is clearer, since we can more readily identify the elements of A. On each set A_s we will define a linear order $<_s$, with the final linear order on A being the limit over s of the orders $<_s$.

A_0 is the set $\{a_0\}$, and $<_0$ is the trivial order on it. At stage $s > 0$ we define

$$c_s = |\{x < s : \varphi_{i,s}(\langle x, \hat{b}\rangle)\downarrow= 1\}|.$$

Thus c_s is the number of predecessors of $\hat{b}$ that have appeared within s steps, and the sequence $\{c_s\}_{s\in\omega}$ is computable and non-decreasing. This is the sequence we will use to determine when C "gives permission" to make changes to $\mathcal{A}$. Also, we define x_s as the greatest length of agreement of C_s with any preceding stage, and t_s as that preceding stage (or the first such stage, if there is more than one), exactly as in Section 2:

$$x_s = \max\{x : (\exists t < s)[x \leq t \;\&\; C_s\upharpoonright x = C_t\upharpoonright x]\},$$

$$t_s = \min\{t : x_s \leq t < s \;\&\; C_s\upharpoonright x_s = C_t\upharpoonright x_s\}.$$

We let $A_s = A_{s-1} \cup \{a_s\}$ and define the order $<_s$ on A_s, considering two cases:

Case A: $c_s > x_s$. We start by ordering $a_0, a_1, \ldots a_{(x_s-1)}$ according to the order $<_{t_s}$. (This is fully defined, since $x_s \leq t_s$.) Preserving the order $<_{t_s}$ on these elements is necessary in order to obey the permitting condition. Since all the remaining elements have subscripts $\geq x_s$, we have permission to move them wherever we like. We place them above $a_0, \ldots a_{(x_s-1)}$, in order by subscript,

$$\underbrace{a_0, \cdots a_{(x_s-1)}}_{\text{in } <_{t_s}\text{-order}} <_s a_{(x_s)} <_s a_{(x_s+1)} <_s \cdots <_s a_s.$$

The idea is that, if we find ourselves in Case A at infinitely many stages, we will build a copy of ω. No new elements will ever be placed to the left of $a_{(x_s)}$ at any stage which lies below s on the approximation-tree, so if s is a true stage, then each of $a_0, a_1, \ldots a_{(x_s-1)}$ will have only finitely many predecessors. We perform this operation when c_s appears to be getting bigger (namely $c_s > x_s$), since this suggests that $\hat{b}$ will have infinitely many predecessors, and thus cannot map to any of $a_0, a_1, \ldots a_{(x_s-1)}$ under any isomorphism of linear orders.

Case B: $c_s \leq x_s$. We preserve the $<_{t_s}$-order on its domain of definition, namely $\{a_j : j \leq t_s\}$, thereby satisfying the permitting condition. Then we insert all new elements, in reverse order of subscript, between the c_s-th and (c_s+1)-st elements of $<_{t_s}$. (Notice that $c_s \leq x_s$ forces $c_s \leq t_s$.) Thus, if we define the subscripts $i_0, \ldots i_{(t_s)}$ so that the $<_{t_s}$-order is given on $a_0, a_1, \ldots a_{t_s}$ by

$$a_{i_0} <_{t_s} a_{i_1} <_{t_s} \cdots <_{t_s} a_{i_{(c_s-1)}} <_{t_s} a_{i_{(c_s)}} <_{t_s} \cdots <_{t_s} a_{i_{(t_s)}},$$

then the new elements are inserted between $a_{i_{(c_s-1)}}$ and $a_{i_{(c_s)}}$,

$$\underbrace{a_{i_0} <_s \cdots <_s a_{i_{(c_s-1)}}}_{\substack{\text{first } c_s \text{ elements} \\ \text{from } <_{t_s}}} <_s \underbrace{a_s <_s a_{s-1} <_s \cdots <_s a_{(t_s+1)}}_{\text{new elements}} <_s \underbrace{a_{i_{(c_s)}} <_s \cdots a_{i_{(t_s)}}}_{\substack{\text{final elements} \\ \text{from } <_{t_s}}}.$$

This is the case where it does not appear that $\hat{b}$ has acquired any new predecessors, so we proceed with the process of building a copy of $c_s + \omega^*$, by inserting new elements immediately after the c_s-th existing element. Each of the first c_s elements under $<_s$ has fewer than c_s predecessors, and by building the ω^*-order above them, we attempt to force every other element of A to have infinitely many predecessors. Our guess at this stage is that $\hat{b}$ has exactly c_s predecessors, and if this guess turns out to be correct, then once again, no isomorphism of linear orders will be able to map $\hat{b}$ to any element of A.

This completes the construction.

Lemma 3.2 (Permitting Condition) *If* $\max(i,j) < m \leq \min(s,t)$ *and* $C_s \restriction m = C_t \restriction m$, *then*

$$a_i <_s a_j \iff a_i <_t a_j.$$

Proof. Assume $t < s$ and induct on s. Since $C_s \restriction m = C_t \restriction m$, we know that $m \leq x_s$. By our construction, $a_i <_s a_j$ if and only if $a_i <_{t_s} a_j$, and by induction, $a_i <_{t_s} a_j$ if and only if $a_i <_t a_j$. ■

Lemma 3.3 *The orders* $<_s$ *converge to a linear order* $<_{\mathcal{A}} = \lim_s <_s$ *on* $A = \bigcup_s A_s (= \omega)$. *Moreover,* $<_{\mathcal{A}}$ *is Turing-computable in* C.

Proof. Given a_i and a_j, find (using a C-oracle) a stage $s > \max(i,j)$ such that $C_s \Restriction \max(i,j) = C \Restriction \max(i,j)$. (Recall that the symbol $S \Restriction x$ denotes $S \restriction (x+1)$.) Now there exists a stage $t_0 > s$ such that for all $t \geq t_0$, $C_t \Restriction \max(i,j) = C \Restriction \max(i,j)$. But then, by the Permitting Condition,

$$a_i <_s a_j \iff (\forall t \geq t_0)[a_i <_t a_j] \iff a_i <_{\mathcal{A}} a_j.$$

Since each $<_s$ is a linear order on A_s, $<_{\mathcal{A}}$ must obey all the axioms for a linear order on A. Moreover, the stage s was computable in C. ■

Notice that the stage s need not be a modulus of convergence (in contrast to the case of c.e. degrees), since there may be a stage $s' > s$ such that $C_{s'} \Restriction \max(j,k) \neq C_s \Restriction \max(j,k)$. We simply know that $<_s$ gives a correct evaluation of the order of a_j and a_k in $\mathcal{A}$.

Proof of Proposition 3.1. We now consider the two statements asserted in Proposition 3.1. First, suppose that $\hat{b}$ has exactly c predecessors in $\mathcal{B}_i$. Let

$\{s_0, s_1, \ldots\}$ be a (noncomputable) enumeration of the true stages in ascending order, and choose k so large that $c_{s_k} = c$ and $x_{s_k} > c$. We write $s = s_k$ to avoid an overabundance of subscripts. Choose subscripts $i_0, i_1, \ldots i_s$ such that the order $<_s$ is given by

$$a_{i_0} <_s a_{i_1} <_s \cdots <_s a_{i_s}.$$

Now Case A will never again apply at any true stage of the approximation, so this order will be preserved at all subsequent true stages. Therefore, at each true stage s_j with $j > k$, the elements $a_{s_{(j-1)}+1}, \ldots a_{s_j}$ are inserted in reverse order of subscript immediately above $a_{i_{(c-1)}}$, as dictated by Case B, with $<_{s_{(j-1)}}$ being preserved on $a_0, a_1, \ldots a_{s_{(j-1)}}$. Thus, if we look only at the true stages, we see the order $c+\omega^*$ being built. But there are infinitely many true stages, so the orders $<_{s_j}$ must converge to $<_{\mathcal{A}}$, and thus $\mathcal{A} \cong c + \omega^*$.

In the other case, when $\hat{b}$ has infinitely many predecessors we claim that $\mathcal{A} \cong \omega$:

Claim 3.4 *If $\hat{b}$ has infinitely many predecessors in $\mathcal{B}_i$, then every element a_x of $\mathcal{A}$ has only finitely many predecessors in A.*

Proof of Claim. As before, let $s_0, s_1, s_2, \ldots$ be the true stages in ascending order, and fix x. Since C is not computable, Lemma 2.3 of Section 2 yields a k so large that $x_{s_k} > x$ and $c_{s_k} > x_{s_k}$. Once again, let $s = s_k$. Let f be the permutation of $\{0, 1, \ldots x_s - 1\}$ such that

$$a_{f(0)} <_s a_{f(1)} <_s \cdots <_s a_{f(x_s-1)}.$$

Pick y such that $f(y) = x$, so a_x has exactly y predecessors under $<_s$.

We claim that for every $j \geq k$, the predecessors of a_x in A_{s_j} are precisely $a_{f(0)}, a_{f(1)}, \ldots a_{f(y-1)}$. For $j = k$ we have the ordering $<_s$ as above on $a_0, \ldots a_{x_s-1}$. Since $c_s > x_s$, we are in Case A of the construction, and all remaining elements are placed above $a_{f(x_s-1)}$, so the only $<_s$-predecessors of a_x are $a_{f(0)}, a_{f(1)}, \ldots a_{f(y-1)}$, as desired. Now assume inductively that these are the only predecessors of a_x under $<_{s_{(j-1)}}$, for $j > k$. Then $<_{s_{(j-1)}}$ is preserved on $a_0, a_1, \ldots a_{(x_{s_j}-1)}$, so by induction, the $<_{s_j}$-predecessors of a_x among these elements are precisely $a_{f(0)}, a_{f(1)}, \ldots a_{f(y-1)}$. If we are in Case A of the construction at stage s_j, then the remaining elements (those with subscripts $\geq x_{s_j}$) are placed above these, yielding no new predecessors to a_x. If we are in Case B, the remaining elements are inserted after the first c_{s_j} of these. But $c_{s_j} \geq c_s$ since $j > k$, and $c_s > x_s > y$, so the new elements are all inserted above a_x, proving the claim. ∎

From Claim 3.4 it is clear that $\mathcal{A} \cong \omega$, independent of the choice of C, as stated in Part 2 of Proposition 3.1. ■

We remark that the Jockusch-Soare basic module in [8] also builds $\mathcal{A} \cong \omega$ whenever $\hat{b}$ has infinitely many predecessors. However, if $\hat{b}$ has exactly c predecessors, it builds $\mathcal{A} \cong d + \omega^*$, for some $d \leq c$, and d varies with the choice of the permitting set C. We avoid that difficulty in Case B of our construction, by placing the new elements between the c_s-th and $(c_s + 1)$-st elements of A_{t_s}. The Jockusch-Soare construction (in their terminology) would place them immediately above the "attached" elements, and the location of the greatest attached element depends on the last permission received, hence depends on C and $\{C_s\}$.

4 Full Construction of the Linear Order

Having seen how this basic module works, we now run it simultaneously for each computable linear ordering $\mathcal{B}_i$. To accomplish this we use the method of separators developed by Jockusch and Soare in [8].

Theorem 4.1 *There exists a linear order $\mathcal{A}$ which has a copy in every non-computable Δ^0_2 degree, but no computable copy. Furthermore, this order may be taken to be of the form*

$$\mathcal{A} = \sum_{i \in \omega} (\mathcal{S}_i + \mathcal{A}_i), \tag{4.1}$$

where each $\mathcal{S}_i \cong 1 + \nu + i + \nu + 1$ and the order type of each $\mathcal{A}_i$ is either ω or $c_i + \omega^* + \omega$ for some $c_i \in \omega$.

(Again ν represents the countable dense linear order with end points.)

We will construct $\mathcal{A}$ by stringing together linear orders $\mathcal{A}_i$, for each $i \in \omega$. The order $\mathcal{A}_i$ is intended to refute the possibility of $\mathcal{A}$ being isomorphic to the linear order $\mathcal{B}_i$ (if any) computed by the i-th computable partial function φ_i. To keep the orders $\mathcal{A}_i$ separate, we insert the computable linear orders $\mathcal{S}_i$ as separators between them. For this we use the notation $\mathcal{C}(\mathcal{A}_0, \mathcal{A}_1, \ldots)$,

$$\mathcal{A} = \mathcal{C}(\mathcal{A}_0, \mathcal{A}_1, \ldots) = \mathcal{S}_0 + \mathcal{A}_0 + \mathcal{S}_1 + \mathcal{A}_1 + \ldots. \tag{4.2}$$

Since no $\mathcal{A}_i$ will have an interval isomorphic to ν, this will enable us to recognize the beginnings and ends of the different $\mathcal{S}_i$'s, and thus to isolate each $\mathcal{A}_i$.

However, the $\mathcal{S}_i$'s cannot be recognized by any computable process. To pick out the first and last points of an $\mathcal{S}_i$, we follow [8] and define Π^0_2 predicates $R_i(e, x_1, \ldots x_{i+6})$ each of which holds just if, in the linear order (if any) determined by φ_e, the points in the separator $\mathcal{S}_i = 1 + \nu + i + \nu + 1$ which are not in the interior of either copy of ν are $x_1, \ldots x_{i+6}$. Then the predicate

$$S_i(x_1, \ldots x_{i+6}, y_1, \ldots y_{i+7}) = R_i(i, x_1, \ldots x_{i+6}) \wedge R_{i+1}(i, y_1, \ldots y_{i+7})$$

is also Π^0_2 and asserts that if φ_i defines a linear order of the form $\mathcal{C}(\mathcal{B}_0, \mathcal{B}_1, \ldots)$, then $x_1, \ldots x_{i+6}$ determine the separator $\mathcal{S}_i$ and $y_1, \ldots y_{i+7}$ determine the separator $\mathcal{S}_{i+1}$. Since the set Inf is Π^0_2-complete, there is a computable function ψ_i whose range is the set ω^{2i+13}, such that for each i and each $\alpha \in \omega^{2i+13}$,

$S_i(\alpha)$ holds if and only if there are infinitely many $s \in \omega$ such that $\alpha = \psi_i(s)$. Moreover, we may choose these functions ψ_i uniformly in i. (In the terminology of [8], ψ_i *assigns chips* to the $(2i+13)$-tuples α, and $S_i(\alpha)$ holds just if α gets infinitely many chips from ψ_i.)

It will be useful for us to assume that the range of ψ_i is all of ω^{2i+13}. If this does not hold for the original ψ_i, we can simply replace it by $\psi_i \oplus \chi_i$, where χ_i is a computable bijection from ω to ω^{2i+13}. The relevant property of ψ_i, namely that $S_i(\alpha)$ holds precisely for those α with $\psi_i^{-1}(\alpha)$ infinite, is clearly preserved under this substitution.

Let $l(\alpha)$ be the $(i+6)$-th element of the $(2i+13)$-tuple α, and $u(\alpha)$ its $(i+7)$-th element. Then α predicts that, if $\mathcal{B}_i$ is of the form $\mathcal{C}(\mathcal{R}_0, \mathcal{R}_1, \ldots)$, the elements of $\mathcal{R}_i$ will be those x such that x lies between $l(\alpha)$ and $u(\alpha)$ in the ordering determined by φ_i, i.e. such that $\varphi_i(\langle l(\alpha), x\rangle) \downarrow = 1 = \varphi_i(\langle x, u(\alpha)\rangle) \downarrow$.

In our construction we will define elements $\hat{b}^s_\alpha$ in the interval $(l(\alpha), u(\alpha))$ of $\mathcal{B}_i$ (where $2i+13 = lh(\alpha)$), which approximate the element $\hat{b}$ from the basic module. (Note that $\hat{b}^s_\alpha$ may be undefined for certain s and α.) Also, if $\hat{b}^s_\alpha$ is defined, we will let

$$c^s_\alpha = |\{x \leq s : \varphi_{i,s}(\langle l(\alpha), x\rangle) \downarrow = 1 = \varphi_{i,s}(\langle x, \hat{b}^s_\alpha\rangle) \downarrow\}|.$$

Thus c^s_α is the number of predecessors of $\hat{b}^s_\alpha$ in the interval between $l(\alpha)$ and $u(\alpha)$, under the order $\mathcal{B}_i$, which have appeared by stage s.

For a given noncomputable Δ^0_2 set C, we now fix i and construct the individual order $\mathcal{A}_i$ as follows (uniformly in i). For each j let $a_j = \langle 2i+1, j\rangle$. (The row $\omega^{[2i]}$ is reserved to form the computable separator $\mathcal{S}_i$, built uniformly in i by a straightforward construction.) The universe A_i of $\mathcal{A}_i$ will be $\omega^{[2i+1]}$, namely $\{a_j : j \in \omega\}$. Thus A_i is computable and infinite. A_i will be the union of sets A^s_α, with α ranging over ω^{2i+13} and $s \in \omega$, and we will write A^s_i for $\bigcup\{A^s_\alpha : \alpha \in \omega^{2i+13}\}$. Each A^s_α is a bin into which we place the elements which we manipulate (at stage s) to try to defeat any possible isomorphism between $\mathcal{A}_i$ and $\mathcal{B}_i$, based on the assumption that $S_i(\alpha)$ holds. (Each element of A_i is used in only one such strategy at stage s, so the different bins at stage s are disjoint: $A^s_\alpha \cap A^s_\beta = \emptyset$ for $\alpha \neq \beta$.)

We now fix i and order the elements α of ω^{2i+13} in order type ω. (Specifically, pick a computable bijection $f_i : \omega^{2i+13} \to \omega$, uniformly in i, and define $\alpha \prec \beta$ if and only if $f_i(\alpha) < f_i(\beta)$.) An α-strategy can only be injured by a β-strategy with $\beta \prec \alpha$, and then only at a stage s such that $\psi_i(s) = \beta$. The strategy which succeeds will be the strategy for that α for which $S_i(\alpha)$ holds, namely the least α such that $\alpha = \psi_i(s)$ for infinitely many s. This strategy will be injured only finitely often by the β-strategies for those $\beta \prec \alpha$, and will not be injured at all by the γ-strategies with $\alpha \prec \gamma$.

The ordering $<_s$ which we define on the elements of A_i^s at stage s will respect the ordering $\prec$, in that for $a_j \in A_\beta^s$ and $a_k \in A_\alpha^s$ with $\beta \prec \alpha$, we will have $a_j <_s a_k$. Also, if $\psi_i(s+1) = \alpha$, the elements from each bin A_γ^s with $\gamma \succ \alpha$ will be taken out of this bin and dumped (all together) into the bin A_α^{s+1} at stage $s+1$. This constitutes an injury to the γ-strategy, which must then start its work anew. We write A_α for the set of elements which reach the α-th bin at some point and stay there forever after,

$$A_\alpha = \bigcup_s \bigcap_{t \geq s} A_\alpha^t.$$

For all $\alpha \in \omega^{2i+13}$, let A_α^0 be the empty set, and let $\hat{b}_\alpha^0$ and c_α^0 be undefined. At each stage $s > 0$, we let $\alpha = \psi_i(s)$.

Step 1. We let

$$A_\alpha^s = \left(\bigcup_{\gamma \succeq \alpha} A_\gamma^{s-1} \right) \cup \{a_s\}.$$

Also, for each $\gamma \succ \alpha$, set $A_\gamma^s = \emptyset$, and for each $\beta \prec \alpha$, set $A_\beta^s = A_\beta^{s-1}$.

Step 2. Let $\hat{b}_\gamma^s$ be undefined for every $\gamma \succ \alpha$, and let $\hat{b}_\beta^s = \hat{b}_\beta^{s-1}$ for every $\beta \prec \alpha$. If $\hat{b}_\alpha^{s-1}$ is defined, let $\hat{b}_\alpha^s = \hat{b}_\alpha^{s-1}$. Otherwise set $n = |\bigcup_{\beta \prec \alpha} A_\beta^s|$, and check whether there are (at least) $n+1$ distinct elements above $l(\alpha)$ and below $u(\alpha)$ in the ordering given by $\varphi_{i,s}$. If so, take $\hat{b}_\alpha^s$ to be the $(n+1)$-st of these, in the ordering given by $\varphi_{i,s}$, so that $c_\alpha^s = n$; if not, then $\hat{b}_\alpha^s$ is undefined.

Step 3. We now define the ordering on A_i^s, by ordering each A_β^s with $\beta \preceq \alpha$ and respecting the order of the bins. As in Section 2, we let

$$x_s = \max\{x : (\exists t < s)[x \leq t \ \& \ C_s \restriction x = C_t \restriction x]\},$$

$$t_s = \min\{t : x_s \leq t < s \ \& \ C_s \restriction x_s = C_t \restriction x_s\}.$$

We will need to preserve the order $<_{t_s}$ on $\{a_j \in A_i^s : j < x_s\}$ in order to obey the permitting condition. Therefore we prove, by induction, that $<_{t_s}$ respects the order of the bins A_β^s. In fact, $<_{t_s}$ respects the order of the bins

A^t_β for every $t > t_s$. The inductive step follows from Step 1, for all j, k, t, β, β', γ, and γ',

$$[a_j \in A^t_\beta \cap A^{t+1}_{\beta'} \ \&\ a_k \in A^t_\gamma \cap A^{t+1}_{\gamma'} \ \&\ \beta \preceq \gamma] \implies \beta' \preceq \gamma'.$$

As in the basic module (see page 11), we now ask, for each $\beta \preceq \alpha$, whether $c^s_\beta > x_s$.

Case A. $c^s_\beta > x_s$, or c^s_β is undefined.

In this case we preserve the order $<_{t_s}$ on $\{a_j \in A^s_\beta : j < x_s\}$. (This will satisfy the permitting condition given below.) Above these elements, but below all elements of $\cup_{\gamma \succ \beta} A^s_\gamma$, we then place all remaining elements of A^s_β, ordered in increasing order of subscript.

Case B. $c^s_\beta \leq x_s$.

In this case we preserve the $<_{t_s}$ order on its entire domain of definition, namely $\{a_j \in A^s_\beta : j \leq t_s\}$. Above these elements we place the elements of $\{a_j \in A^s_\beta : j > t_s \ \&\ \psi_i(j) \succ \beta\}$, in increasing order of subscript. We then put the elements of $\{a_j \in A^s_\beta : j > t_s \ \&\ \psi_i(j) = \beta\}$ in reverse order of subscript and place them consecutively so that the leftmost of them is the $(c^s_\beta + 1)$-st element of $\bigcup_{\beta' \preceq \beta} A^s_{\beta'}$. (If there are fewer than c^s_β elements in $\cup_{\beta' \preceq \beta} A^s_{\beta'}$ already ordered by $<_s$, then we simply put these new elements at the right end of A^s_β, again in reverse order of subscript.) This completes the construction.

The ordering $\mathcal{A}$ which is the goal of this paper will be precisely

$$\mathcal{C}(\mathcal{A}_0, \mathcal{A}_1, \ldots) = \mathcal{S}_0 + \mathcal{A}_0 + \mathcal{S}_1 + \mathcal{A}_1 + \ldots \quad .$$

Notice that since the entire construction was uniform in i, we can string the $\mathcal{S}_i$'s and $\mathcal{A}_i$'s together computably. We show below that $\deg(\mathcal{A}_i) \leq_T C$ for each i, so $\mathcal{A}$ will be Turing-reducible to C. (The orders $\mathcal{S}_i$ are all computable, uniformly in i.) Indeed, the $\mathcal{S}_i$ and $\mathcal{A}_i$ were constructed so that the union of all their universes is precisely ω. The ordering $<_\mathcal{A}$ respects the rows of ω, and within each row $\omega^{[2i]}$ or $\omega^{[2i+1]}$ it is given by the ordering on $\mathcal{S}_i$ or $\mathcal{A}_i$, respectively.

The proofs of the following two lemmas are identical to those of Lemmas 3.2 and 3.3 in the basic module.

Lemma 4.2 (Permitting Condition) *If $C_s \restriction m = C_t \restriction m$ and $a_j, a_k \in A_i$ with $j, k < m \leq \min(s, t)$, then*

$$a_j <_s a_k \text{ if and only if } a_j <_t a_k.$$

∎

Lemma 4.3 *For each i, the orders $<_s$ converge to a linear order $<_{\mathcal{A}_i}$ on $A_i = \bigcup_s A_i^s$ $(= \omega^{[2i+1]})$. Moreover, $<_{\mathcal{A}_i}$ is Turing-computable in C, uniformly in i.* ∎

Lemma 4.4 *For any two noncomputable Δ_2^0 sets C and C', any computable approximations $\{C_s\}$ and $\{C'_s\}$, and any i, the linear orders $\mathcal{A}_i$ and $\mathcal{A}'_i$ built by the above construction are isomorphic.*

Proof. We will show that each order $\mathcal{A}_i$ built by the construction is independent of C. Notice that the only time C is used in the construction is in Step 3, and there it rearranges the order of certain elements but never moves elements from one A_α^s to another A_β^s. The movement of elements from one A_α^s to another A_β^s depends only on the function ψ_i. Therefore, for each α and s, the set A_α^s is independent of C, although the ordering of the elements of the set may depend on C. Also, the definitions of the elements $\hat{b}_\alpha^s$ in Step 2 depend only on φ_i, ψ_i, and the sizes of the sets A_α^s, all of which are independent of C.

Fix i, and let $\alpha \in \omega^{2i+13}$ be minimal such that $\psi_i^{-1}(\alpha)$ is infinite. (If $\psi_i^{-1}(\alpha)$ is finite for all α, then every A_α is finite, so $\mathcal{A}_i \cong \omega$, independent of choice of C.) Let $s_0, s_1, \ldots$ be the true stages in the approximation $\{C_s\}$ of C, in increasing order.

We deal first with the case in which $\lim_s \hat{b}_\alpha^s$ diverges. Pick the least true stage s_q such that $\psi_i(s) \succeq \alpha$ for all $s \geq s_q$. By Step 2 of the construction, we know that if $s \geq s_q$ and $\hat{b}_\alpha^s$ is defined, then $\hat{b}_\alpha^{s+1}$ is defined and equals $\hat{b}_\alpha^s$. Therefore, $\hat{b}_\alpha^s$ must be undefined for every $s \geq s_q$. But then every corresponding c_α^s is undefined, so in Step 3 after stage s_q, we always are in Case A, which instructs us simply to place the elements with subscripts $\geq x_s$ at the right end of A_α^s, in increasing order of subscript. Finitely many elements lie in $\cup_{\beta \prec \alpha} A_\beta$, and any other element a_j must wind up in A_α. (Initially a_j may go into some A_γ^t with $\gamma \succ \alpha$, but it will be dumped into $A_\alpha^{t'}$, at the next t' with $\psi_i(t') = \alpha$.) Eventually we will reach a true stage s_p with $a_j \in A_\alpha^{s_p}$ and $j < x_{s_p}$, and at all true stages thereafter, no more elements will be placed below a_j. Since the orders $<_s$ converge and the true stages form an infinite subsequence, this means that a_j can have only finitely many predecessors in $<_{\mathcal{A}_i}$. So the order $\mathcal{A}_i$ is isomorphic to ω, independent of choice of C.

Now suppose that the elements $\hat{b}_\alpha^s$ converge to some element $\hat{b}_\alpha$ of $\mathcal{B}_i$. Then the sequence $\{c_\alpha^s\}$ is defined for cofinitely many s and either converges to some $c_\alpha \in \omega$ (if $\hat{b}_\alpha$ has exactly c_α predecessors in the interval $(l(\alpha), u(\alpha))$ of $\mathcal{B}_i$) or goes to infinity (if $\hat{b}_\alpha$ has infinitely many predecessors there).

In the case with only finitely many predecessors, we choose a true stage $s = s_q$ so large that $c^s_\alpha = c_\alpha$ and $x_s > c_\alpha$ and $\psi_i(t) \succeq \alpha$ for all $t \geq s$. Then for each true stage s_p with $p > q$, we have $c^{s_p}_\alpha = c_\alpha < x_s \leq x_{s_p}$ so we are in Case B of Step 3 of the construction. Therefore, at each such s_p, we preserve $<_{s_{(p-1)}}$ on its domain of definition, $A_i^{s(p-1)}$. Define the numbers $i_0, i_1, \ldots i_s \in \{0, 1, \ldots s\}$ so that

$$a_{i_0} <_s a_{i_1} <_s \cdots <_s a_{i_s}.$$

Since $t_{s_p} = s_{(p-1)}$, induction on p yields

$$a_{i_0} <_{s_p} a_{i_1} <_{s_p} \cdots <_{s_p} a_{i_s}.$$

Moreover, since we are in Case B at every such true stage, no element is ever inserted to the left of the c_α-th element $a_{i_{(c_\alpha - 1)}}$. Thus the order which we build will have initial segment c_α.

We claim that the rest of the order has type $\omega^* + \omega$, so that the entire order has type $c_\alpha + \omega^* + \omega$. The ω^*-chain is built of those elements a_j with $j > s$ and $\psi_i(j) = \alpha$. There are infinitely many such elements still to be added to A_i, and each of them, once added, will be inserted (possibly along with other elements) immediately after $a_{i_{(c_\alpha - 1)}}$ at the next true stage, building the ω^*-chain above $a_{i_{(c_\alpha - 1)}}$.

The ω-chain is built of those elements a_j with $j > s$ and $\psi_i(j) \succ \alpha$. (There are infinitely many such, since the range of ψ_i is all of ω^{2i+13}.) For such an element, let t be the first stage such that $a_j \in A^t_\alpha$, and let s_p be the first true stage $\geq t$. If there is no true stage between stage j and stage t, then a_j will be placed (possibly along with other elements) at the right end of $A^{s_p}_\alpha$, by Case B of Step 3. If there was a true stage between j and t, then a_j will be placed at the right end of $A^{s_p}_\alpha$ (possibly along with other elements) by the preservation of the order $<_{s_{(p-1)}}$ at stage s_p. In either case, $t_{s_{(p+1)}} = s_p \geq j$, and since we are in Case B at every true stage after s, the order $<_{s_p}$ is preserved (on its domain of definition) at every subsequent true stage. New elements a_k will be added at subsequent true stages only to the right of a_j (if $\psi_i(k) \succ \alpha$) or immediately after $a_{i_{(c_\alpha - 1)}}$ (if $\psi_i(k) = \alpha$). Since the true stages form an infinite subsequence, this allows us to deduce the type of the order $\mathcal{A}_i$: it will be of the form $c_\alpha + \omega^* + \omega$. Thus the order type of $\mathcal{A}_i$ is independent of C in this case.

In the case where the interval $(l(\alpha), \hat{b}_\alpha)$ of $\mathcal{B}_i$ is infinite, we claim that $\mathcal{A}_i \cong \omega$.

Claim 4.5 *If* $\lim_s c_\alpha^s = \infty$*, then each* $a_j \in A_\alpha$ *has only finitely many predecessors in* $\mathcal{A}_i$*.*

Proof. Fix j. There will be a true stage $s = s_q$ for which $x_s > j$ and $(\forall t \geq s)\psi_i(t) \succeq \alpha$, and by Lemma 2.3, we may also assume that $c_\alpha^s > x_s$. Therefore, at stage s we will be in Case A of Step 3, so all elements a_k of A_α^s with $k \geq x_s$ will be placed above the elements of $\{a_m \in A_i^s : m < x_s\}$, and hence above a_j. Thus a_j has fewer than x_s predecessors under $<_s$, and all of those predecessors have subscripts $< x_s$ and therefore will precede a_j at every subsequent true stage s_p.

We now induct on the true stages s_p with $p > q$, to see that the predecessors of a_j under each $<_{s_p}$ are precisely the predecessors of a_j under $<_{s_{(p-1)}}$. Let s_p be a true stage with $p > q$. If we are in Case B of Step 3 at stage s_p, then the ordering $<_{s_{(p-1)}}$ is not injured, and all new elements are placed either after the $c_\alpha^{s_p}$-th element, hence to the right of a_j (since $c_\alpha^{s_p} \geq c_\alpha^s > x_s$ and by induction j has fewer than x_s predecessors under $<_{s_{(p-1)}}$), or else at the right end of $A_\alpha^{s_p}$. Thus a_j receives no new predecessors at such a stage. If we are in Case A of Step 3 at stage s_p, then all elements with subscripts $\geq x_{s_p}$ are moved to the right end of $A_\alpha^{s_p}$, and all other elements, including a_j and all its predecessors, are left alone. Therefore, for each $p > q$, the predecessors of a_j under $<_{s_p}$ are precisely the predecessors of a_j under $<_s$. Since the true stages form an infinite subsequence of ω, we see that indeed a_j has only those (finitely many) predecessors under $<_{\mathcal{A}_i}$, just as we had claimed. ∎

This holds for every $a_j \in A_\alpha$, while each A_β $(\beta \prec \alpha)$ is finite and each A_γ $(\gamma \succ \alpha)$ is empty, so clearly $\mathcal{A}_i \cong \omega$, independent of the choice of C. (Notice that we did use the noncomputability of C in applying Lemma 2.3.) This completes the proof of Lemma 4.4. ∎

Corollary 4.6 *For each* i*, the linear order* $\mathcal{A} = \mathcal{C}(\mathcal{A}_0, \mathcal{A}_1, \ldots)$ *has a unique interval isomorphic to* $\mathcal{S}_i$*.*

(Here $\mathcal{C}$ is the operator defined in Equation (4.2), so $\mathcal{A}$ is precisely the order given in (4.1).)

Proof. From the proof of Lemma 4.4, we see that the only possible outcomes of the construction of each $\mathcal{A}_i$ are ω and $n+\omega^*+\omega$, where n is finite. None of these has an interval isomorphic to ν, the countable dense linear order with end points, but every one is infinite, so the only copy of $1+\nu+i+\nu+1$ in $\mathcal{A}$ is $\mathcal{S}_i$ itself. ∎

Corollary 4.7 *$\mathcal{A}$ is not isomorphic to any of the computable linear orders $\mathcal{B}_i$.*

Proof. We note first, using the preceding corollary, that if $\mathcal{A} \cong \mathcal{B}_i$ for some i, then $\mathcal{B}_i$ has unique intervals isomorphic to $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$. Hence there is a unique $\alpha \in \omega^{2i+13}$ for which $S_i(\alpha)$ holds, so $\psi_i^{-1}(\alpha)$ is infinite, but $\psi_i^{-1}(\beta)$ is finite for all $\beta \neq \alpha$. Since $\mathcal{A} \cong \mathcal{B}_i$, $\mathcal{A}_i$ must be isomorphic to the interval $(l(\alpha), u(\alpha))$ of $\mathcal{B}_i$.

If the sequence $\langle \hat{b}^s_\alpha \rangle$ diverges, then $\hat{b}^s_\alpha$ is undefined for cofinitely many s, as noted in the proof of Lemma 4.4. By Step 2 of the construction, this can only happen if the interval $(l(\alpha), u(\alpha))$ contains at most $|\bigcup_{\beta \prec \alpha} A_\beta|$ elements, But $\mathcal{A}_i \cong \omega$, so $\mathcal{B}_i \not\cong \mathcal{A}$.

If $\hat{b}^s_\alpha$ converges to an element $\hat{b}_\alpha$ with only c_α-many elements between $l(\alpha)$ and $\hat{b}_\alpha$, then $\mathcal{A}_i \cong\ c_\alpha + \omega^* + \omega$. Thus every element of $\mathcal{A}_i$ has either fewer than c_α predecessors or infinitely many in $\mathcal{A}_i$, so no isomorphism could take $\hat{b}_\alpha$ to any element of $\mathcal{A}_i$.

Finally, if $\hat{b}^s_\alpha$ converges to an element $\hat{b}_\alpha$ with infinitely many elements between $l(\alpha)$ and $\hat{b}_\alpha$, then $\mathcal{A}_i \cong \omega$, so again there can be no isomorphism taking $\hat{b}_\alpha$ to any element of $\mathcal{A}_i$. ∎

Thus $\mathcal{A}$ is a linear order with no computable copy. However, for every noncomputable Δ^0_2 set C, we have seen (in Lemma 4.3) that there is a copy of $\mathcal{A}$ computable in C. We discuss Julia Knight's full theorem (from [9]) in the next section, as Theorem 5.2, but an easy consequence of it, cited in [8] and [3], implies that for each such C, there is a copy of $\mathcal{A}$ whose Turing degree is exactly the degree of C. This is precisely the property we had promised would hold for $\mathcal{A}$.

5 Further Questions

The obvious generalization of Theorem 4.1 would be a positive answer to Downey's third question:

Question 1.4 (Downey) *Is there a linear order whose spectrum contains every degree except* $\mathbf{0}$*?*

This question remains open, however. It is known that for every noncomputable degree $\mathbf{C}$ there is a linear order whose spectrum includes $\mathbf{C}$ but not $\mathbf{0}$. However, Knight's proof of this result (see [3]) is highly nonuniform: one uses the Downey-Seetapun result for Δ_2^0 degrees, a coding construction for non-low$_2$ degrees, and a combination of these two techniques for the remaining degrees. Therefore, it would be far harder to make Knight's construction yield the same result independent of the choice of $\mathbf{C}$, as we managed to do for the Jockusch-Soare construction.

A more general question, also posed by Downey [3], is simply to ask what spectra are possible for a linear order.

Question 5.1 (Downey) *What can be said about Spec*($\mathcal{L}$) *for a given linear order* $\mathcal{L}$*?*

There are two main results so far. One we have already used in proving Theorem 4.1, namely Knight's result that the spectrum must be closed upwards under Turing reducibility. This follows from a stronger theorem of Knight [9].

Theorem 5.2 (Knight) *If* $\mathcal{A}$ *is any structure, then exactly one of the following two statements holds:*

(5.1) For all Turing degrees $\mathbf{C} <_{\mathbf{T}} \mathbf{D}$*, if there is an isomorphic copy of* $\mathcal{A}$ *of degree* $\mathbf{C}$*, then there is an isomorphic copy of* $\mathcal{A}$ *of degree* $\mathbf{D}$*;*

(5.2) There exists a finite subset S *in the universe* A *of* $\mathcal{A}$ *such that any permutation of* A *fixing* S *is an automorphism of* $\mathcal{A}$*.*

For any infinite linear order $\mathcal{L}$, (5.2) clearly fails, so the upward-closure property (5.1) holds. (If $\mathcal{L}$ is finite, then (5.2) holds, and indeed in this case every copy of $\mathcal{L}$ is computable.)

The second main result about the spectrum of a linear order is due to Richter [12]:

Theorem 5.3 (Richter) *If the spectrum of a linear order has a least degree, then that degree is* $\mathbf{0}$.

The least degree of the spectrum of a structure is often simply called the degree of the isomorphism type of that structure. Thus, Richter's result says that $\mathbf{0}$ is the only possible degree for the isomorphism type of a linear order; a linear order with no computable copy cannot have any least degree in its spectrum. This can be viewed as a result on the difficulty of coding sets into linear orders. If we wish to code a noncomputable set S into a linear order, so that S would be computable from every copy of the order, then that order cannot have a copy computable from S. (Otherwise, $\deg(S)$ would be the least degree of the spectrum of the linear order.)

These two results rule out many possible spectra for linear orders. On the other hand, Theorem 4.1 is an example of a positive response to Question 5.1: the spectrum can contain all Δ^0_2 degrees except $\mathbf{0}$. We can also use Knight's result on noncomputable degrees to show that it is possible to separate any two degrees $\mathbf{C} <_{\mathbf{T}} \mathbf{D}$ via the spectrum of a linear order. That is:

Corollary 5.4 *If* $\mathbf{C} <_{\mathbf{T}} \mathbf{D}$*, then there exists a linear order* $\mathcal{L}$ *such that* $\mathbf{D} \in Spec(\mathcal{L})$ *and* $\mathbf{C} \notin Spec(\mathcal{L})$.

Proof. Simply take Knight's proof for the case $\mathbf{C} = \mathbf{0}$ and relativize it to the degree $\mathbf{C}$. ■

We might ask if it is possible to separate any two Turing degrees in this way, even if they are incomparable. Also, we can ask if it is possible to separate collections of degrees:

Question 5.5 *If* $\mathbf{P}$ *and* $\mathbf{N}$ *are collections of Turing degrees such that no degree in* $\mathbf{P}$ *is reducible to any degree in* $\mathbf{N}$*, is there a linear order* $\mathcal{L}$ *whose spectrum contains all of* $\mathbf{P}$ *but does not intersect* $\mathbf{N}$*?*

This question is intended to be asked for specific choices of $\mathbf{P}$ and $\mathbf{N}$, particularly classes of c.e. sets (or Δ^0_2 sets) whose indices cannot be computably separated. We have seen in the preceding sections that linear orders can contain more information than subsets of integers. There is no set which is computable in every nonzero Δ^0_2 degree but not in $\mathbf{0}$, whereas there is a linear order which is computable in every Δ^0_2 degree except $\mathbf{0}$. What else can linear orders do? For instance, could a linear order contain enough information to separate the high Δ^0_2 sets from the low ones?

Clearly the answer to Question 5.5 is not always positive, for otherwise we could contradict Richter's result by taking **P** to be the upper cone above a noncomputable set C, including the degree of C itself, and **N** to be the complement of **P**. Thus no linear order characterizes the ability to compute C, whereas the set C itself does. Here, then, is an example in which a set contains information which a linear order cannot contain. Knight's and Richter's results both clearly restrict the amount of information encoded in a linear order. Perhaps there are other common mathematical structures which escape Richter's restriction, which would entail failing her "Recursive Enumerability Condition" (see [12]). Knight's restriction appears inevitable, since under (5.2) in Theorem 5.2, the information contained by the structure is essentially encoded in a single finite set.

References

[1] C. J. Ash, C. G. Jockusch & J. F. Knight; Jumps of Orderings, *Transactions of the American Mathematical Society* **319** (1990), 573-599.

[2] J. C. E. Dekker; A Theorem on Hypersimple Sets, *Proceedings of the American Mathematical Society* **5** (1954), 791-796.

[3] R. Downey; On Presentations of Algebraic Structures, in *Complexity, Logic, and Recursion Theory*, ed. A. Sorbi (New York: Dekker, 1997), 157-205.

[4] R. Downey; Recursion Theory and Linear Orderings, in *Handbook of Computable Algebra* [to appear].

[5] R. Downey & C. G. Jockusch; Every Low Boolean Algebra Is Isomorphic to a Recursive One, *Proceedings of the American Mathematical Society* **122** (1994), 871-880.

[6] R. Downey & J. F. Knight; Orderings with α-th Jump Degree $\mathbf{0}^{(\alpha)}$, *Proceedings of the American Mathematical Society* **114** (1992), 545-552.

[7] L. J. Feiner; *Orderings and Boolean Algebras Not Isomorphic to Recursive Ones*, PhD. Thesis, MIT (1967).

[8] C. G. Jockusch & R. I. Soare; Degrees of Orderings Not Isomorphic to Recursive Linear Orderings, *Annals of Pure and Applied Logic* **52** (1991), 39-64.

[9] J. F. Knight; Degrees Coded into Jumps of Orderings, *Journal of Symbolic Logic* **51** (1986), 1034-1042.

[10] J. F. Knight & M. Stob; Computable Boolean Algebras, *Journal of Symbolic Logic* [to appear].

[11] D. Posner; A Survey of Non-R. E. Degrees $\leq \mathbf{0}'$, in *Recursion Theory: Its Generalizations and Applications*, ed. F. R. Drake & S. S. Wainer (Cambridge: Cambridge University Press, 1980), 52-109.

[12] L. J. Richter; Degrees of Structures, *Journal of Symbolic Logic* **46** (1981), 723-731.

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

[14] T. Slaman; Relative to any Nonrecursive Set, *Proceedings of the American Mathematical Society* **126** (1998), 2117-2122.

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

[16] J. Thurber; *Degrees of Boolean Algebras*, PhD. Thesis, University of Notre Dame (1994).

[17] S. Wehner; Enumerations, Countable Structures, and Turing Degrees, *Proceedings of the American Mathematical Society* **126** (1998), 2131-2139.

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF CHICAGO
CHICAGO, ILLINOIS 60637
E-mail: russell@math.uchicago.edu