RECURSION THEORY AND LINEAR ORDERINGS

Rod Downey
Mathematics Department
Victoria University
P. O. Box 600
Wellington, New Zealand

June 11, 2005

e-mail: DOWNEY@MATH.VUW.AC.NZ

Author's research partially supported by grants from the IGC at Victoria University, by the MSRI at Berkeley and by the US/NZ cooperative science foundation. He wishes to thank Jeff Remmel and Bob Soare for helpful discussions. He wishes to thank Peter Donelan for much needed help with the overall presentation, and to thank Carl Jockusch and Michael Moses who supplied extensive corrections and revisions.

1 Introduction

The goal of this article is to give a survey of results and techniques in a fascinating area of algorithmic combinatorics. Computability theory, or recursion theory[1], is concerned with our attempts to understand the algorithmic nature (or lack thereof) of the universe; combinatorics - particularly finite combinatorics - is concerned with counting, sorting and matching arguments. These areas combine particularly well, and the resulting theory is quite deep.

The main idea behind this area of computable combinatorics is the following. When we deal with 'real' structures in everyday mathematics, we would deal with objects such as the integers whose existence is somehow concrete. Similarly if we argue that a certain feature exists - say a solution to an equation - it is desirable to be able to give an algorithmic proof and hence *exhibit* a solution. In computable mathematics, this intuition can be formalised by asking that our structures be given in some algorithmic manner, and then examining the algorithmic content of such structures. In a loose sense one can view this as a theory of interactive programming where one feeds in an algorithm for the structure, and asks whether one can derive algorithms for various features of the structure, or else prove that this is impossible. Classical instances of such questions are the word problem of groups, Hilbert's tenth problem, the decidability of the elementary theory of abelian groups, and the four-colour problem. In the word problem we are given a group by a 'finite presentation' i.e. $G = \langle x_1, \ldots, x_n | y_1, \ldots, y_m \rangle$ where $\langle x_1, \ldots, x_n \rangle$ is a free group and $y_1, \ldots, y_m$ generate a free normal subgroup of $\langle x_1, \ldots, x_n \rangle$. One then asks if there is an algorithm to decide if a word in $x_1, \ldots, x_n$ is the identity or not in G. A famous result of Novikov [**?**] and Boone [24] shows that it is possible to select G so that the answer is no. Hilbert's tenth problem asks whether we can decide if a given polynomial (in many variables) over $\mathbb{N}$ has positive integer solutions. Again the answer is no (Matijasevic [**?**]). The question on abelian groups is whether given a simple formula $\varphi(x_1, \ldots, x_n)$ in the language of groups, we can decide if $\varphi(x_1, \ldots, x_n)$ is true of all abelian groups. Here the answer is that an algorithm does indeed exist (W. Szmieliew [218]). Finally we know by Appel and Haken that any finite planar graph is 4-colourable. The question is how fast can this be done? Recently Appel and Haken have found a fast (polynomial time) algorithm to do this.

In our studies we seek to understand the relative difficulty of such problems.

At this point we should stress that we are concerned with studying these problems in the area of classical mathematics and not as 'constructivists' whose philosophical outlook is to regard as not existing anything which cannot be 'theoretically' actually constructed. As we shall see, there are also some related questions in so-called 'reverse mathematics'. This area seeks to classify problems proof-theoretically (according to how much 'comprehension' is needed) rather than algorithmically. (see eg Friedman-Simpson-Smith [71]).

In the present paper, we shall study the computability theory of linear orderings. There is, of course, a lot of other work on computable combinatorics which we shall not examine (although we do give some brief comments in Section 9). Bill Gasarch has a survey article devoted to this area in the present volume.

Not counting the introduction the paper is divided into eight sections.

[1]The study of the effective content of mathematics has been referred to in many ways. Folowing early work of Kleene it has been known as "recursion theory." However, we prefer to use the name "computability theory" which captures the original spirit and intentional meaning of the area. We refer the reader to Soare [**?**] for a discussion of these issues.

We hoped to make the paper accessible to a (reasonably determined) combinatorialist who was only noddingly acquainted with computability theory. Thus in Section 2 we give a brief account of the basics of computability theory needed for this article. Some proofs are sketched to give the reader a flavour of the types of arguments we need. This section might also serve as a reminder to those logicians who work in areas other than computability theory. Experts should jump immediately to Section 3.

In Section 3 we analyse the notion of presentation of a structure (ordering). This generalises the notion of presentation of a group (alluded to above) and is crucial to the remainder of the paper. The key tension is between the notions of *effective* invariant and *classical* invariant. Here we investigate results such as those of Chisholm and Moses [31], Feiner [63, 64], Rosenstein [195], Richter [188]. Understanding the degrees of presentability has been an area of intensive research and we refer the reader to Downey [**?**] for further details here.

In Section 4 we consider how computable properties may partition classical isomorphism types of computable orderings. We begin by leading up to Remmel's [182] characterisation of computablely categorical linear orderings as those with finitely many successivities. We then turn to the results of Moses [157] and Schwartz [202] on more restrictive classes such as 1-computable and decidable orderings.

In Section 5 we introduce the infinite injury priority method and the fundamental results of Watnick [225] and Lerman [130]. We give a previously unpublished proof (due to the author) of Watnick's [225] result on finite condensations of order types. For those familiar with the standard 'tree-of-strategy' Π_2 arguments, it is hoped that this will be much more accessible than either Watnick's [225] or Ash-Jockusch-Knight's [8] version. (No knowledge of Π_2 arguments is assumed here. The main thrust of this chapter is trying to understand the complexity of distinguished suborderings of an ordering. For instance, Watnick's theorem asserts that τ is an order type with a Π_2 copy iff $\zeta\tau$ has a computable copy. Here ζ denotes the order type of the integers). We also present in this section some related work of Fellner [62], Rosenstein, Moses and the author.

In Section 6 we look at the effective content of various classical theorems, concentrating on embeddings and automorphisms. Here, by effective content we mean the following. Consider the result that every infinite linear ordering has an infinite ω or ω^* sequence (here ω denotes the order type of $\mathbb{N}$ and ω^* its reversal). We ask is this true effectively? That is, does each *effective* infinite linear ordering have an *effective* ω or ω^* sequence? An old theorem of Tennenbaum [unpubl.] and independently Denisov [unpubl.] says the answer is no in a very strong way. So in this section we concentrate on extensions of the Tennenbaum-Denisov result by Watnick, Hird, Lerman and others, and other related questions on automorphisms and distinguished subsequences. Of course, if a theorem fails to hold effectively, we seek effective versions. For instance, as we see in this section, Rosenstein showed that if R is a computable ordering then R has a computable subsequence of order type $\omega, \omega^*, \omega+\omega^*$ or $\omega+\zeta\eta+\omega^*$ (here η denotes the order type of the rationals) and Lerman proved that all of these types are necessary. Another effective version of the same theorem is Manaster's result that R must have a Π_1 ω or ω^* sequence. Thus there can be many effective versions of the same theorem. We also look at the effective content of the Dushnik-Miller theorem on self embeddings, and finally look at the Downey - Moses and Schwartz results on automorphisms and self embeddings.

In Section 7 we look at linear extensions of computable partial orderings and so the effective content of Szpilrajn's [214] result that any *poset has a linear extension*; which leads to the well-known theory of dimension for posets. Here we also analyse the Slaman-Woodin result of dense linear extensions. We look at Kierstead's beautiful results on computable covering numbers for Dilworth's theorem and the theory of computable dimension developed by Kierstead, Rosenstein, McNulty, Schmerl, Trotter and others. Finally we also look at the Manaster - Rosenstein - Remmel theory of computable 2-dimensional partial orderings. The results of this section are certainly not well known by computability theorists and really deserve to be, as they form a very attractive theory with many difficult open questions.

In Section 8 we study degree theoretical aspects of presentations, and arguments of the 'worker types' whose roots go back to work of Harrington on Peano arithmetic. For instance we look at the Jockusch-Soare[101] result that there are low linear orderings not isomorphic to computable ones and give a new proof of the Downey-Moses [53] result on degrees of successivities in linear orderings. We finish this section with the Ash-Jockusch-Knight [8] and Downey-Knight [50] results on jump degrees of orderings. (The arguments here are computability-theoretically very sophisticated and are mainly only very briefly sketched.)

In the last section (Section 9) we briefly discuss other areas of computable combinatorics, and other areas of computable aspects of linear orderings.

The philosophy of the paper is to give at least one example of all of the techniques of proof. There are a couple of exceptions where, for brevity, we have omitted a result that would take us too far afield (eg Knight's proof that if A has proper 1-degree then that degree is $\mathbf{0}'$, which uses forcing) and in other cases we have been quite sketchy in the interests of keeping this paper of finite length and concentrating on the *ideas*. In the case of the Kierstead - McNulty - Trotter - Rosenstein results in Section 7, there is also a very nice survey by Kierstead [113] and McNulty [145] that the reader is urged to read. (We have tried to avoid the game-theoretical language used by these authors and this makes our treatment rather different from either of these surveys.)

Finally we remark that there are quite a few new results, unpublished results and new proofs of old results included here, and numerous open questions scattered throughout. In the case of open questions, we have tried to find out exactly to whom they were due but in many cases they seem to belong to folklore.

Notation is standard. We let $\omega, \omega^*, \zeta, \eta$ denote the order types of, respectively, $\mathbb{N}, -\mathbb{N}, \mathbb{Z}$, and $\mathbb{Q}$. An order type is called *scattered* if it does not embed η as a subordering. We define the standard sum $\alpha + \beta$ and product $\alpha \cdot \beta$ of orderings via: $\alpha + \beta$ results from putting a copy of α followed by one of β, and $\alpha \cdot \beta$ results from replacing each point in an ordering of type β by one of type α. We say that (x, y) is called a *successivity* or *adjacency* in an ordering A if $(\forall z \in A)(z \notin (x, y))$. The *width* of a poset is the size of the largest antichain. We let $\langle , \rangle$ denote a standard bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$. We let $2^{<w}$ denote the tree of all binary sequences and 2^w the collection of all paths through $2^{<w}$. Any other needed definitions or terminology will be introduced in the paper.

2 Preliminaries

Many mathematicians unfortunately do not have a training in recursion theory. Hopefully this section will give a reasonably self-contained account of some of the basics in this area. For more details the reader is referred to, for example, Salomaa [201], Rogers [194], Soare [213], or Davis and Weyuker [40].

Our concern is with functions from $A \to B$ where $A, B \subseteq \mathbb{N}$; i.e. *partial* functions on $\mathbb{N}$. If $A = \mathbb{N}$ with the function is called *total.* Looking only at $\mathbb{N}$ may seem rather restrictive, but remember we are dealing with algorithmic structures, and all such structures (from our point of view) can be coded as subsets of $\mathbb{N}$. For example, if we consider the rationals Q, these can be considered as coded in $\mathbb{N}$ as follows:

> Let $r \in \mathbb{Q} - \{0\}$; write $r = (-1)^\delta \left(\frac{p}{q}\right)$ with $p, q \in \mathbb{N}$ in lowest terms and $\delta = 0$ or 1. Then define the Gödel number of r, $\#(r)$, as $2^\delta 3^p 5^q$, with the Gödel number of 0 to be 0.

Then by the fundamental theorem of arithmetic, $\#$ describes an injection from $\mathbb{Q}$ into $\mathbb{N}$ and furthermore given $n \in \mathbb{N}$ we can decide exactly which $r \in \mathbb{Q}$, if any, has $\#(r) = n$. Similarly if σ is a finite binary string, say $\sigma = a_1, a_2, \ldots a_n$ then we can define $\#(\sigma) = 2^{a_1+1}3^{a_2+1}\ldots(P_n)^{a_n+1}$ where P_n denotes the n-th prime.

The above procedures are called 'effective coding' since they give an algorithm for the relevant injection.

Speaking of algorithms, what exactly do we mean? Here we return to the celebrated Church-Turing Thesis. Paraphrased in modern terminology, the Church-Turing Thesis states that *the collection of algorithmic partial functions are exactly those that can be computed by Pascal programmes.* Of course, neither Turing nor Church said it like this. Rather, there are many formulations of the class of computable functions - via Turing machines, λ-calculus, Kleene schemes, etc, but *they all turn out to be equivalent.* We will call them the *partial recursive functions or the computable partial functions.* If a partial recursive function is total we shall simply call it *recursive or computable.* For our purposes, it is really not necessary to describe them in detail, but we will need to know two very important properties of this class of functions:

Property 2.1 (Enumeration Theorem-Universal Turing Machine) *There is an alg or computableorithmic way of enumerating all the computable partial functions. That is, there is a computable partial function $f(x, y)$ of two variables such that*

$$f(x, y) = \varphi_x(y)$$

where $\varphi_x(y)$ denotes the x-th computable partial function on input y; and this makes sense.

Sketch Proof To see that property 2.1 holds, again we can return to Gödel numbering. A programme is just a finite set of lines with a finite set of possible statements on each line (more or less). We can assign to statements such as '`else`', '`go to`' etc Gödel numbers, so that each line could be assigned a number. For example, if `GO TO` were assigned to 50,

```
15 GO TO 25
```

could be numbered as

$$3^{\#(15)} \; 5^{\#(50)} \; 7^{\#(25)}.$$

We do this so that each line has a unique number and then code up the programme as $2^{\#(l_1)}3^{\#(l_2)} \ldots (P_n)^{\#(l_1)}$ where $\#(l_i)$ denotes the Gödel number of l_i.

In this way we can clearly establish property 2.1. $\square$

The *point* of property 2.1 is that we can *pretend* that we have all the machines $\varphi_1, \varphi_2 \ldots$ in front of us; to compute 10 steps in the computation of the 3rd machine on input 20, we can pretend to walk to the 3rd machine, put 20 on the tape and run it for 10 steps (we write this as $\varphi_3^{10}(20)$). In many ways, property 2.1 is the platform that makes undecidability proofs work since it allows us to diagonalize *over* the class of computable partial functions *without leaving the class.* For instance

Property 2.2 (Unsolvability of the Halting problem) *There is no algorithm which given x, y decides if $\varphi_x(y)\downarrow$, that is, if the x-th programme on input y halts. Indeed there is no algorithm to decide if $\varphi_x(x)\downarrow$.*

Proof Suppose such an algorithm exists. Then by property 2.1, it follows that the following function g is (total) computable:

$$g(x) = \begin{cases} 1 & \text{if } \varphi_x(x)\uparrow \text{ (i.e. } \varphi_x(x) \text{ does not halt)} \\ \varphi_x(x)+1 & \text{if } \varphi_x(x)\downarrow \end{cases}$$

Again using property 2.1, there is a y with $g = \varphi_y$. As $g(y)\downarrow$, $\varphi_y(y)\downarrow$ and hence $g(y) = \varphi_y(y) + 1 = g(y) + 1$, a contradiction. $\square$

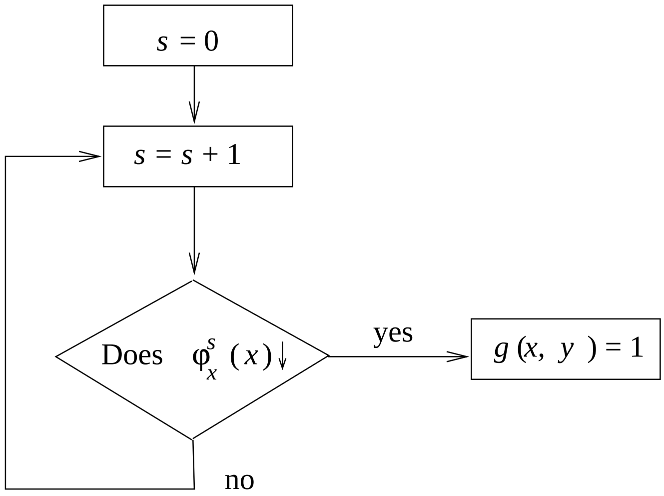
Note that we *can* define a computable *partial* g via $g(x) = \varphi_x(x)+1$ and avoid contradiction as it will follow that, for any *index* y of g, $\varphi_y(y) = g(y)\uparrow$. (In fact such a g has no computable (total) extension (exercise).) Also the reason for the use of *partial* computable functions in property 2.1 is clear. The argument above will show that there is no computable procedure to generate all (and only) the total computable functions. Actually the result above can be used to show many problems are algorithmically unsolvable by 'coding' the halting problem. For example

Proposition 2.3 *There is no algorithm to decide whether dom φ_x is empty.*

Proof We code the halting problem into the problem of deciding if dom $\varphi_x = \emptyset$. That is we show that *if* we could decide whether dom $\varphi_x = \emptyset$ *then* we could solve the halting problem. Define a computable partial function via, for all $y \in \mathbb{N}_0$,

$$g(x, y) = \begin{cases} 1 & \text{if } \varphi_x(x)\downarrow \\ \uparrow & \text{if } \varphi_x(x)\uparrow. \end{cases}$$

To see that g is computable partial, compute $g(x, y)$ using the flowchart below.



Now we can consider $g(x, y)$ as a computable collection of computable partial functions (this is called the *s-m-n theorem*). The *s-m-n* theorem asserts there is a computable $s(x)$ such that, for all x, y

$$\varphi_{s(x)}(y) = g(x, y).$$

Then

$$\mathrm{dom}(\varphi_{s(x)}) = \begin{cases} \mathbb{N} & \text{if } \varphi_x(x)\downarrow \\ \emptyset & \text{if } \varphi_x(x)\uparrow. \end{cases}$$

So if we could decide for a given x if $\varphi_{s(x)}$ has empty domain, we could solve the halting problem. □

The reasoning used in proposition 2.3 can be pushed a lot further. First we can regard problems as coded by subsets of $\mathbb{N}$. For example, the halting problem is coded by $K = \{x : \varphi_x(x)\downarrow\}$ (or, for example, $K^* = \{2^x 3^y : \varphi_x(y)\downarrow\}$). We need some terminology.

Definition 2.4 A set $A \in \mathbb{N}$ is called

i) *computably enumerable* (r.e) if $A = \mathrm{dom}\varphi_x$ for some x.
ii) *computable* if A and $\bar{A}(= \mathbb{N} - A)$ are both computably enumerable.

We will let W_e denote the e-th computably enumerable set. That is, we let $W_e = \mathrm{dom}\varphi_e$ and let $W_{e,s} = \{x \leq s \,|\, \varphi_e^s(x)\downarrow\}$ constitute s steps in the enumeration of W_e.

The name *computably enumerable* comes from a notion of 'effectively countable' via the following characterisation.

Proposition 2.5 *An infinite set A is computably enumerable iff there is a (total) computable injective function f with $f(\mathbb{N}) = A$.*

Proof ($\Rightarrow$) Suppose $A = \mathrm{dom}\varphi_x$. Compute f in stages. To compute $f(0)$, find the least s and then the least z for s such that $z \leq s$ and $\varphi_x^s(z)\downarrow$. Define $f(0) = z$. To compute $f(1)$ find the least $s_1 \geq s$ and the least $z \neq z_1 \leq s_1$ such that $\varphi_x^{s_1}(z_1)\downarrow$. $f(s_1) = z_1$, and similarly we define $f(j)$ at step j to the the least element of $\mathrm{dom}\varphi_x^{s_j}$ not yet in the range of f. Clearly f is computable and injective.

($\Leftarrow$) Let $f(\mathbb{N}) = A$ as in the hypothesis of the theorem. Define $g^s(x) = 1$ iff $f(s) = x$. Then g is clearly computable partial and dom $g = ra\ f = A$. $\square$

Thus we can think of an computably enumerable set as an effective infinite list. (But **not** *computably in order.*) Note that computable sets correspond to decidable questions; for if A is computable, then to decide if $x \in A$, let $f(\mathbb{N}) = A$ and $g(\mathbb{N}) = \bar{A}$. Now enumerate $f(0)$, $g(0)$, $f(1)$, $g(1)$, $\ldots$ until x occurs (as it must) in the A or $\bar{A}$ list. Note also that property 2 states that K= $\{x : \varphi_x(x) \downarrow\}$ is an computably enumerable but not computable set.

An *index set* is a set A such that if $x \in A$ and $\varphi_x = \varphi_y$ then $y \in A$. Most of the sets we have looked at are index sets (e.g. $\{x : \text{dom}\varphi_x = \emptyset\}$ is an index set). Generalising proposition 2.3 we have.

Theorem 2.6 (Rice) *An index set A is computable (and so the problem it codes is decidable) iff $A = \mathbb{N}$ or $A = \emptyset$.*

Proof The proof of Rice's Theorem is very similar to the proof of 2.3.

Let $A \neq \mathbb{N}$, $\emptyset$ be an index set and without loss of generality, $e \in A$ where $\text{dom}\varphi_e = \emptyset$. Fix $z \in A$. Then for some q, $\varphi_z(q) \downarrow$. By the technique of proposition 1 (*s-m-n* theorem), there is a computable $s(x)$ such that, for all $y \in \mathbb{N}$,

$$\varphi_{s(x)}(y) = \begin{cases} \varphi_z(y) & \text{if } \varphi_x(x) \downarrow \\ \uparrow & \text{if } \varphi_x(x) \uparrow \end{cases}.$$

Then $\varphi_x(x) \downarrow$ implies $\varphi_{s(x)} = \varphi_z$ and so $s(x) \in A$, and $\varphi_x(x) \uparrow$ implies $\varphi_{s(x)} = \varphi_e$ and so $s(x) \notin A$. So if A were computable, K would also be computable. $\square$

Of course many decision problems are not coded by index sets and so can have decidable solutions.

The idea used in these proofs is that of *reducibility.* Namely, we use the strategy of "if we can do A then it allows we to do B". We can formalise this as follows.

We can regard one problem as being at least as hard as another by attaching to our computer an infinite read-only memory. We then say $A \leq_T B$ (A is Turing reducible to B) if we can compute A given that B is put in order into the memory, and we are allowed to ask for each x, a finite number of questions of the memory during the computation that decides whether a particular x is in A. This is a partial ordering. In the example above (proposition 2.3) let $E = \{x : \text{dom}\varphi_x = \emptyset\}$, we showed that $K \leq_T E$ by showing that $x \in K$ iff $s(x) \in E$. Indeed Rice's theorem demonstrates that a nontrivial index set I will always have $K \leq_T I$.We will write $A \equiv_T B$ if $A \leq_T B$ and $B \leq_T A$. The equivalence classes give a notion of equicomputability and are called *degrees (of unsolvability).* If $A \leq_T B$ we say A is *B-computable.* We let $\mathbf{0}$ denote the degree of the computable sets. We call machines with the infinite memory attached *oracle machines.* We can regard normal machines as oracle machines with empty memory.

We get the analogue of propositions 2.3 and 2.5 for oracle machines. That is

Property 2.7 *There is an enumeration of all oracle machines* $\{\Phi_x : x \in \mathbb{N}\}$

Notation 2.8 We will let $\Phi_{e,s}(C;x)$ denote s stages in the computation of Φ_e with oracle C and input x.

Property 2.9 *If A is any set and $\Phi_x(A;y)$ denotes the result, if any, of computing the result of input y on the x-th oracle machine with oracle A (i.e. A in the memory), then $K^A = \{x : \Phi_x(A;x)\downarrow\}$ is not A-computable.*

We refer to K^A as 'K relativized to A', and the process of generalising a fact about computably enumerable sets to all oracles as *relativization.* Note that $A <_T K^A$ (see below).

Remark 2.10 *As we will see, we can also put resource bounds on our procedures. For example if we count steps and ask that computations halt in a polynomial (in the length of the input) number of steps we arrive at the polynomial computable functions, and the notion of polynomial time reducibility. We will discuss this more fully in Section 9.*

If A is a set, K^A is a standard operator on A called the jump operator. We write A' for K^A, if the degree of A is **a** then we write **a**' for $\deg(A')$ (we use boldface letters for degrees). Since $B'T > B$ for all B, consequently we have defined a hierarchy of degrees

$$\mathbf{0}, \mathbf{0}', \mathbf{0}'', \ldots$$

If $\mathbf{a} < \mathbf{0}^{(}\mathbf{n})$ for some n we call **a** *arithmetical.* Note that since each degree is countable and has only countably many predecessors (only countably many machines) there are uncountably many degrees but only countably many arithmetical ones.

The name *arithmetical* comes from the following alternative methods of defining (e.g.) $\mathbf{0}'$. We define the notions of Σ_n, Π_n and Δ_n as follows:

We say a computable set is Δ_1. A set B is called Σ_n if $x \in B$ holds iff there is a computable relation $R(x_1, \ldots, x_n, x)$ with

$$\underbrace{\exists x_1\ \forall x_2\ \exists \ldots}_{n \text{ alternations of quantifiers}} \qquad R(x_1, \ldots, x_n, x)$$

We similarly say that a set Q is Π_n as we did for B except we now have the leading quantifier $\forall$ followed by n alternations of quantifiers. Note that we can always collapse two quantities of the form $\exists x_1\ \exists x_2$ into a single one of the form $\exists x_3$ using Gödel numbers which is why it makes sense to only look at *alternations* of quantifiers. Finally we say a set R is Δ_n if it is both Σ_n and Π_n. With these notions we get the arithmetical hierarchy of Kleene

$$\begin{tikzcd}[every arrow/.append style={dash}]
 & \Pi_1 \arrow[dr] & & \Pi_2 \arrow[dr] & & {} & \\
\Delta_1 \arrow[ur] \arrow[dr] & & \Delta_2 \arrow[ur] \arrow[dr] & & \Delta_3 \arrow[ur] \arrow[dr] & & \cdots \\
 & \Sigma_1 \arrow[ur] & & \Sigma_2 \arrow[ur] & & {} &
\end{tikzcd}$$

Here lines mean inclusion (rightward along the page) and all inclusions are proper. There is a strong relationship between the above and degrees. For instance

Theorem 2.11 (Kleene) *A set A is computably enumerable iff A is Σ_1.*

Proof Suppose A is computably enumerable Then $A = \mathrm{dom}\varphi_x$ for some x. Then $y \in A$ iff $(\exists s)(\varphi_x^s(y)\downarrow)$. Conversely if A is Σ_1 then for some computable R we have $y \in A$ iff $(\exists z)(R(z,y))$. Define a computable partial g by setting $g(y) = 1$ at stage s iff $s \geq y$ and $\exists z < s$ with $R(z,y)$ holding. $\square$

In terms of degree, $K = \emptyset'$ is the most complicated computably enumerable set. To see this, let $K_0 = \{\langle x,y\rangle : x \in W_y\}$ where $\langle , \rangle$ is a computable bijection from $\mathbb{N}\times\mathbb{N}$ to $\mathbb{N}$. Note that if W is computably enumerable then $W \leq_T K_0$ since $x \in W$ iff $\langle x,\, y\rangle \in K_0$ where $W = W_y$. As an example of a typical reduction, we show that $K_0 \equiv_T K$. First $K \leq_T K_0$ since K is computably enumerable Now to see that $K_0 \leq_T K$, use the s-m-n theorem to define a computable class of functions:

$$\varphi^{(z)}_{s(<x,\,y>)} = \begin{cases} 1 & \text{if } \varphi_x(y)\downarrow \text{(i.e. } x \in W_y) \\ \uparrow & \text{otherwise} \end{cases} \tag{1}$$

Then $\langle x,\, y\rangle \in K_0$ iff $\varphi_x(y)\downarrow$ iff $\varphi_{s(<x,y>)}(s\langle x,\, y\rangle))\downarrow$ iff $s(\langle x,\, y\rangle) \in K$. Hence $K_0 \leq_T K$.

It can be shown that a set $A \leq_T K$ iff A is Δ_2 and, more generally $A \leq_T \emptyset^{(n)}$ iff A is Δ_{n+1}. The arithmetical sets are those you 'get from Peano arithmetic' (roughly speaking).

To finish this section we will discuss a very important problem in the degrees: Post's problem. By the above we see that all index sets are of degree $\geq \mathbf{0}'$. Post [174] observed that in 1944 all known computably enumerable problems had the property that they were either of degree $\mathbf{0}$ or $\mathbf{0}'$. He asked

Question 2.12 *Post's Problem Does there exist an computably enumerable degree $\mathbf{a}$ with $\mathbf{0} < \mathbf{a} < \mathbf{0}'$?*

Post's problem was finally solved by Friedberg [67] and Muchnik [162] using a new and ingenious method called the priority method that we shall meet in Section 3. They showed that

Theorem 2.13 (Friedberg, Muchnik) *There exist computably enumerable degrees $\mathbf{a}|\mathbf{b}$. That is $\mathbf{a} \not\leq \mathbf{b}$ and $\mathbf{b} \not\leq \mathbf{a}$.*

There have been many extensions of the Friedberg - Muchnik theorem. Some notable extensions due to Sacks [200] : the computably enumerable degrees are dense, and Lachlan [125] and Yates [226] : there exist computably enumerable degrees $\mathbf{a},\ \mathbf{b} \neq \mathbf{0}$ with $\mathbf{a} \cap \mathbf{b} = \mathbf{0}$ and $\mathbf{c}, \mathbf{d}$ with $\mathbf{c} \cap \mathbf{d}$ not existing. Thus the computably enumerable degrees are a dense upper semilattice that is not a lattice and is not homogeneous. Soare [213] contains a lot of information on the computably enumerable degrees, computably enumerable sets and related structures.

We will not prove any of these results, and will look at the priority method later. There is however one crucial ingredient of the proof of the Friedberg - Muchnik theorem that we will use in some of the work to follow and which we will look at in this section.

Lemma 2.14 (Use principle) *Suppose $\Phi(A;\, x)\downarrow$. Let $u(\Phi(A;\, x))$ denote the maximum element x such that the question "is x in the memory?" is called during the computation of $\Phi(A;\, x)$. Let B be any set such that $B[u] = A[u]$ where for any set C, $C[z] = \{p:\ p \le z$ and $p \in C\}$ and $u = u(\Phi(A;\, x))$. Then $\Phi(A;\, x) = \Phi(B;\, x)$.*

Proof Both A and B give the same answers to the question in the computations, hence the result must be the same. □

One use of Lemma 2.14 is an important characterisation of Δ_2 sets (i.e. sets $A \le_T K$).

Lemma 2.15 (Shoenfield Limit Lemma) *A is Δ_2 iff there is a computable function $g(\,,\,)$ such that*

(i) $\lim_s g(x, s)$ *exists (i.e.* $|\{s:\ g(x, s) \neq g(x, s+1)\}| < \infty$*)*

(ii) $\lim_s g(x, s) = A(x)$ *where we identify sets with their characteristic functions. (That is, $A(x) = 0$ if $x \notin A$ and $A(x) = 1$ if $x \in A$.)*

Proof ($\Rightarrow$) Suppose A is Δ_2. That is $A \le_T \emptyset'$ so that for some procedure Φ_e, we have $\Phi_e(K) = A$. Define $g(x, s) = 0$ if $\Phi_{e,s}(K_s;\, x)\uparrow$ or $\Phi_{e,s}(K_s;\, x)\downarrow\neq 1$ and $g(x, s) = 1$ otherwise.

(Here $K = \cup_s K_s$ is some computable enumeration of K.) Given x, let $u = u(\Phi_e(K;\, x))$. Let $s = s(x)$ be the stage where $K_s[u] = K[u]$. Let $t \ge s$ be the least stage such that $\Phi_{e,t}(K;\, x)\downarrow$. As $K_t[u] = K_s[u] = K[u]$, we have, by the use principle, $\Phi_{e,t}(K;\, x) = \Phi_{e,t}(K_t;\, x) = \Phi_{e,q}(K_q; x) = \Phi_e(K;\, x)$ for all $q \ge t$. Then for all $q \ge t$, $g(x, t) = g(x, q) = A(x)$ by definition.

($\Leftarrow$) Suppose such a function g exists. We construct an computably enumerable set B and a reduction Γ so that $\Gamma(B) = A$. Then $A \le_T \emptyset'$ since A is computably enumerable (and so $A \le_T \emptyset'$ by the fact that $K_0 \le_T K$ in (1)).

We define $\Gamma(B;\, x)$ and B in stages, for all x.

Stage 0 Let $\gamma(x, 0) = \langle x,\, 0\rangle$ and define $\Gamma_0(B_0;\, x) = g(x,\, 0)$ and $B_0 = \emptyset$.

Stage $s + 1$ We have defined $\gamma(x,\, s)$. If $g(x,\, s+1) = g(x,\, s)$ keep $\gamma(x,\, s) = \gamma(x,\, s+1)$, $B_{s+1} = B_s$ and $\Gamma_{s+1}(B_{s+1};\, x) = \Gamma_s(B_s\, :\, x)$. If $g(x,\, s+1) \neq g(x,\, s)$ enumerate $\gamma(x,\, s)$ into B_{s+1}. Set $\Gamma_{s+1}(B_{s+1};\, x) = g(x,\, s+1)$. It will be the case that $\gamma(x,\, s) = \langle x,\, j\rangle$ for some $j \in \mathbb{N}$. Set $\gamma(x,\, s+1) = \langle x,\, j+1\rangle$.

The construction succeeds as we now see. Note that $\lim_s \gamma(x,\, s) = \gamma(x)$ exists since $\gamma(x,\, s{+}1) \neq \gamma(x,\, s)$ iff $\gamma(x,\, s+1) \neq \gamma(x,\, s)$. Furthermore $\gamma(x,\, s+1) \neq \gamma(x,\, s)$ iff $\gamma(x,\, s) \in B_{s+1} - B_s$.

Also, $\lim_s \Gamma_s(B_s;\, x) = \Gamma(B;\, x)$ since $\Gamma_s(B_s\, :\, x) \neq \Gamma_{s+1}(B_{s+1};\, x)$ iff $\gamma(x,\, s) \in B_{s+1} - B_s$. It follows that Γ is B-computable. B can decide $\Gamma(B;\, x)(= A(x))$ as follows. Given x and B, compute the least s such that $\gamma(x,\, s) \notin B$. Then $\Gamma_s(B_s;\, x) = \Gamma(B;\, x) = A(x) = g(x, s) = g(x)$ by construction. □

Remark Intuitively in the ($\Leftarrow$) part of the above proof K can answer $(\exists s)(g(x,\, s) \neq g(x,\, s{+}1))$ and hence computably in K we can compute the limit of $g(x,\, s)$ and hence compute $A(x)$.

3 Presentations

Before we go any further we need to clarify what we mean by presenting a structure. First, all universes are identified with $\mathbb{N}$ under some coding, and hence we need only worry about the complexity of the ordering relation. We will write $\mathbf{A} = (\mathbb{N}, \leq_A)$ for an ordering and will write $\leq$ for $\leq_A$ when the context is clear.

Definition 3.1 A linear ordering $\mathbf{A} = (\mathbb{N}, \leq_A)$ is called *computably presented* if $\leq_A$ is a computably relation (on $\mathbb{N}$). [Equivalently, the open diagram)or a *computably ordering*) of $\mathbf{A}$ is computable.]

'Effectivising' the proof often attributed to Cantor, we see:

Theorem 3.2 $\mathbf{A}$*is computably presented iff* $\mathbf{A}$*is computably isomorphic to a computably subset of* $\mathbb{Q}$ *(under the usual ordering).* We call this a computably subordering of $\mathbb{Q}$.

Proof Fix some Gödel numbering of $\mathbb{Q}$ which we denote by $\#$. Label $\mathbf{A}$ as $\{a_0, a_1, \ldots\}$. At stage $s+1$, we will have already defined $f(a_0), \ldots, f(a_s)$. Now find the appropriate i and/or j such that $a_i < a_{s+1} < a_j$ and for all $k \leq s$, if $k \neq i, j$ then $a_k \notin (a_i, a_j)$. We can compute a rational r_{s+1} with $\#(r_{s+1}) > s+1$, and $f(a_i) < r_{s+1} < f(a_j)$. Define $f(a_{s+1}) = r_{s+1}$.

It is easy to see that f is an embedding, and furthermore raf is a computably set (as $raf = B$ is increasing so $x \in B$ iff x is a member of B by stage (x)). $\square$.

For the purposes of our investigations we'd like to extend Definition 3.1 and Theorem 3.2. We say A is a Σ_n - (resp. Δ_n-, Π_n-, $\mathbf{a}$-) *presented* linear ordering if $\leq$ is a Σ_n (resp. Δ_n-, Π_n-, degree $\mathbf{a}$-) set.

If we relativise Theorem 3.2 we see that if $\mathbf{A}$ is $\mathbf{a}$-presented, then $\mathbf{A}$ is isomorphic to a subordering of $\mathbb{Q}$ of degree$\mathbf{a}$. Suppose that $\mathbf{A}$ is Σ_1-presented. We claim that $\mathbf{A}$ is isomorphic to a Π_1 subordering of $\mathbb{Q}$. If $\mathbf{A}$ is Σ_1 then $x \leq y$ iff $(\exists z)(R(x, y, z)$ holds. Since $\leq$ is linear we know at least one of $x \leq y$ or $y \leq x$ hold. By searching, given x and y we can decide one of $x \leq y$ or $y \leq x$ holding. The problem is that we can't decide if $x = y$ holds (i.e. if **both** hold). Note that $x = y$ is Σ_1, so there is a computably relation $\widehat{R}$ such that $x = y$ iff $\exists s\ \widehat{R}(x, t, s)$. Write $x \neq_s y$ if $\neg\widehat{R}(x, y, s)$.

Now follow the proof of Theorem 3.2. In 3.2, at stage s, we need to work out all the $a_i \leq a_j$ relationships for $i, j < s$. Now, if we follow 3.2, the problem we need to resolve is that we see $a_i \leq a_j$ and $a_i \neq_s a_j$ so we define $f_s(a_i) = r_1$ and $f_s(a_j) = r_2$, say, with $r_1 \neq r_2$. Now at some stage $t > s$ it may be that $a_i =_t a_j$ (and hence for all $q \geq t$, $a_i =_q a_j$). Let $i < j$, then our action is to delete r_2 from raf by defining $f_t(a_i) = f_t(a_j) = r_1$. Choosing $i < j$ ensures that $(\forall i)(\lim_s f_s(a_i)$ exists) and by extensions of embeddings, f is a monomorphism. We have thus proven:

Theorem 3.3 *If* $\mathbf{A}$ *is* Σ_1*-presented, then* $\mathbf{A}$ *is* Δ_2*-isomorphic to a* Π_1 *subordering of* $\mathbb{Q}$.

Relativizations and similar arguments give:

Theorem 3.4 (Feiner [63], essentially) *(i) If* $\mathbf{A}$ *is* Σ_n*-presented then* $\mathbf{A}$ *is* Δ_{n+1}*-isomorphic to a* Π_n *subordering of* $\mathbb{Q}$*.*

(ii) If $\mathbf{A}$ *is* Π_n*-presented then* $\mathbf{A}$ *is* Δ_{n+1}*-isomorphic to a* Σ_n *subordering of* $\mathbb{Q}$*.*

And hence

(iii) If $\mathbf{A}$ *is* Δ_n*-presented then* $\mathbf{A}$ *is* Δ_n*-isomorphic to a* Δ_n *subordering of* $\mathbb{Q}$*.*

We remark that Theorem 3.4 can be extended to all the computably ordinals using either a 'worker argument' (see Section 9) or some other scheme using the uniformities involved at limit stages.

Thus we can (more or less) restrict ourselves to studying suborderings of $\mathbb{Q}$ (certainly up to Turing degree).

Definition 3.5 Henceforth we shall call a linear ordering Π_n ($\Sigma_n,\ \Delta_n$ etc) if it is a Π_n ($\Sigma_n,\ \Delta_n$) subordering of $\mathbb{Q}$.

Note It is important that the reader note that a Σ_n*-presented linear ordering* is different from a Σ_n *linear ordering.* Indeed, a Σ_n-presented ordering is isomorphic to a Π_n linear ordering by Theorem 3.4.

Using a very similar argument to that used in Theorem 3.2 it is possible to show that a Σ_1 linear ordering is isomorphic to a computably linear ordering. By relativization this means that a Σ_n linear ordering is isomorphic to a Δ_n linear ordering. By the limit lemma, we know that if A is a Δ_2 subset of $\mathbb{Q}$ then there is an approximation $g(\,,\,)$ such that $g(x,\,s) \neq g(x,\,s+1)$ only finitely often, and with $\lim_s g(x,\,s) = A(x)$. Using this fact, again we can use a modification of Theorem 3.2 to show that a Δ_n linear ordering is isomorphic to a Π_1 linear ordering. By relativization it follows that a Δ_{n+1} linear ordering is isomorphic to a Π_n linear ordering, and hence *a* Σ_{n+1} *linear ordering is isomorphic to a* Π_n *linear ordering.* We now give an alternative proof of this result.

Theorem 3.6 (Feiner [63]) *If* $\mathbf{A}$ *is a* Σ_{n+1} *linear ordering the* $\mathbf{A}$ *is isomorphic to a* Π_n *linear ordering.*

Proof (Rosenstein [195, p.431]) Recursively partition $\mathbb{Q}$ into half open intervals $\{I(r) :\ r \in \mathbb{Q}\}$ so that if $r <_{\mathbb{Q}} s$ then $I(r)$ is all left of $I(s)$, and so that given q we can compute the unique r, if any, with $q \in I(r)$. We can suppose there is a computably function p such that $\{p(x,\,r) :\ x \in \mathbb{N}\}$ enumerates $I(r)$. Now let A be a Σ_{n+1} linear ordering. It is not difficult to show *there is a* Π_n *predicate* R *such that* $r \in A \leftrightarrow (\exists x)R(x,\,r)$ *and furthermore if* $r \in A$ *there is a unique* x *with* $R(x,\,r)$. Define B via $q \in B \leftrightarrow (q = p(x,\,r)$ and $R(x,\,r))$. Then B is Π_n and evidently $B \cong A$. $\square$

It follows that the arithmetical hierarchy of order types for subordering of $\mathbb{Q}$ looks like

$$\Delta_1 \subseteq \Pi_1 \subseteq \Pi_2 \subseteq \Pi_3 \cdots \tag{2}$$

and again continue through the computably ordinals.

We now show that all the inclusions of (2) are proper.

Theorem 3.7 (Feiner [63, 64]) *For all n, there exists a Π_{n+1} linear ordering not isomorphic to a Π_n linear ordering and a Π_1 linear ordering not isomorphic to a computably one.*

Proof Theorem 3.4, it suffices to construct a Σ_1-presented linear ordering not isomorphic to a computably one, and then relativise the result.

Given an ordering $\mathbf{A}$, define the (n-) **block relation** as $B(n)$: there is a set of n elements $x_1 \ \dots \ x_n$ with $x_1 < x_2 < \cdots < x_n$ such that $(x_i,\, x_{i+1})$ is a *successivity* (i.e. $(\neg\exists y)(x_i < y < x_{i+1})$) and for no y is $(y,\, x)$ or $(x_n,\, y)$ a successivity.

Then it is easy to see that $\quad B(A) = \{n : \ B(n) \text{ holds } \}$ is Σ_3^A.

[The relevant definition is

$n \in B(A)$ iff $(\exists x_1, \dots, x_n)(\forall y)(\exists z)((x_1 < \cdots < x_n)$ and $\bigwedge_{i=1}^{n-1} \neg(x_i < y < x_{i+1})$ and $(y < x_1 \rightarrow y < z < x_1)$ and $(x_n < y \rightarrow x_n < z < y))$.]

The reader should note that $B(A)$ is a classical invariant.

In particular if $\mathbf{A}$ is computably the $B(\mathbf{A})$ is Σ_3 but if $\mathbf{A}$ is Σ_1-presented and hence Δ_2-presented, then $B(\mathbf{A})$ is apparently only Σ_4 (since to assert (eg) $x_1 < x_2$ we need to say that $(\forall s)(x_1 <_s x_2)$ of $\mathbf{A}$ is Σ_1-presented).

In view of Theorem 3.4, it suffices to show that, given any set A we can construct an A-computably linear ordering $\mathbf{A}$ wwith $B(\mathbf{A})$ a set that is Σ_3^A complete (i.e. of degree A'''). Let D be any Σ_3^A set. If D is Σ_3^A then for some A-computably relation R we have

$$n \in D \text{ iff } (\exists x)(\forall y)(\exists z) R(x,\, y,\, z,\, n).$$

The ordering $\mathbf{A}$ we construct will have order type

$$\zeta + n_1 + \zeta + n_2 + \zeta + n_3 + \cdots$$

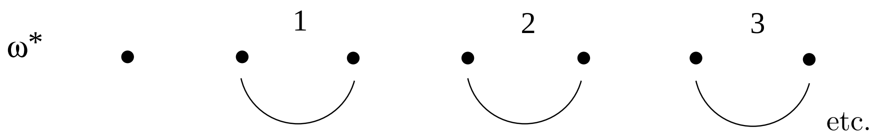
where the reader will recall ζ denotes the order type of the integers $\mathbb{Z}$ and we will use the n_i to represent D via $B(\mathbf{A})$. Since we can relativise to arbitrary A, we might as well take $A = \emptyset$ so D is Σ_3.

We begin the construction with ω copies of ζ and construct a computably ordering of order type $\omega.\zeta$. The x-th of the ω copies of η is labelled x. Fix x. We regard this ζ as being split as

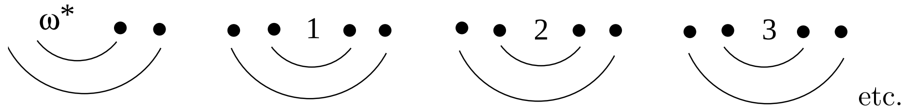
$$\omega^* + 1 + 3 + 3 + 4 + 5 + \cdots$$

During the construction, we will add new points to the left and right of some of the blocks 1, 2, 3, ... above. Below we will have a notion of n being infinitively often "verified". This verification machinery will select certain n to be 'built around'. *Without* this machinery, the idea is simple. At stage s, put a point on each side the block we are building around each original n block (as well as at the end of the ω^* sequence). So at stage s we will have made the n block into a $s + n + s$ block.)

So at stage 1 we'd put



At stage 2 we'd get:



Thus, *without* the verification machinery, at the end we'd get a sequence of ω blocks of type $\zeta = \omega^* + \omega$.

Now at stage s of the construction, we say $(n,\, x)$ *is verified (for y)* if

(i) for all $z < y$, $(n,\, x)$ is verified for z.

(ii) $(n,\, x)$ is not yet verified for y.

(iii) $R(x,\, y,\, t,\, n)$ holds for some $t < s$.

At stage s, for each n, if $(n,\, x)$ becomes verified for some y, we define $(n,\, x)$ as now verified and then in the x-th copy of ζ we also add a point *immediately* to the left and to the right of n. (i.e. the original distinguished n-block). So, for instance, if $(2,\, x)$ were verified for all y but neither $(3,\, x)$ nor $(1,\, x)$ were, then at the end of the construction we'd have the x-th copy of ζ looking like:

$$\omega^* + \omega + 2 + \omega^* + \omega + \cdots$$

as the 1 and 3 would be absorbed into the $\omega^* + \omega$'s. Now as $n \in D \leftrightarrow (\exists x)(\forall y)(\forall z)R(x,\, y,\, z,\, n)$ then we see that the only n which survive are those in S (namely those for which there is an x such that $(n,\, x)$ is verified infinitely often). Hence $B(\mathbf{A}) = D$.

To finish the proof we need only choose some $A \leq_T \emptyset'$ with $\Sigma_3^A \not\leq_T \emptyset'$. This is a standard result from classical computability theory. Indeed for any A there is a Σ_3^A-complete S (so that $D \equiv_T A'''$).

Remarks 3.8 (i) Feiner [63, 64] also proved that there were Σ_1-presented *boolean algebras* not isomorphic to computably ones. This a much harder argument involving simultaneously coding $\mathbf{0}^{(n)}$ for all n. Since every computable boolean algebra can be represented as the boolean algebra of left closed right open intervals of a computable linear ordering, this result implies the linear ordering one. Thses ideas have been pushed further by Remmel, Goncharov, Thurber, Jockusch and Soare and others. We refer the reader to Downey [**?**] for further details.

(ii) Define a set to be *low* if $A' \equiv_T \emptyset'$. It is known that there are low non-computably sets. The argument above shows that there are Π_1 linear orderings not isomorphic to low linear orderings. To a certain extent this is a drawback of the coding techniques used, since it is not clear if there are low linear orderings not isomorphic to a computably one. There are, as we shall see later.

(iii) Feiner's techniques were quite novel at the time since nobody had ever coded anything except computably enumerable non-computably sets previously. The first significant simplification and systemization of Feiner's theorem was in John Thurber's Thesis [**?**] and also can be found in Downey [**?**].

(iv) For those interested in classical computability theory, the above results allowed Feiner to:

  (a) show that the lattice of computably enumerable sets $\mathcal{E}$ is not computably presentable (using Lachlan's results on boolean algebras and hh-simple sets) and

  (b) solve the strong homogeneity conjecture. This asked if, in the language of degrees with jump, whether the cones above any two degrees were isomorphic. The answer is no, using the above and initial segment results. (See Feiner [63, 64].) (Nowadays we use distributive lattices to get this, see Lerman [131]

(v) It is not difficult to completely characterise the computably presentable well orderings: indeed if A is any arithmetical subset of $\mathbb{Q}$ that is well ordered, then the order type of A is a computably ordinal and hence computably presentable. (Rice [187] and see Rosenstein [195, Theorem 16.37]. For related results we refer the reader to Pinus [173], Hay, Manaster and Rosenstein [87], Watnick [224] and Rice [187].)

One might wonder as to what complexity we might give an *order type*, rather than a specific *presentation.* One suggestion of Jockusch was: if $\mathbf{A}$ is a structure then the *degree of the isomorphism type* of $\mathbf{A}$ is the minimum of the degrees of all the presentations of $\mathbf{A}$. Unfortunately this is not a useful definition for linear orderings.

Theorem 3.9 (Richter [188]) *If* $\mathbf{A}$ *is an order type with a degree, then that degree is* $\mathbf{0}$*. In fact, if* A *is a given subordering of* $\mathbb{Q}$*, there is a* $B \cong A$ *such that the degrees of* A *and* B *has infimum* $\mathbf{0}$.

Proof The argument is an easy diagonalization of a type called the finite extension technique. Suppose A is a linear ordering not isomorphic to a computably one. We build $B \cong A$ with B a subordering of $\mathbb{Q}$ via an isomorphism $f : B \mapsto A$ in stages such that the degree of B and the degree of A has infimum $\mathbf{0}$. To do this we need to meet the requirements (for $e \in \mathbb{N}$):

$$R_{\langle e,\, i\rangle} : \ \Phi_e(B) = \Phi_i(A) = g \quad \Rightarrow \quad g \text{ is computable.}$$

At stage $s+1$ we attend to R_s. Let $s = \langle e,\, i\rangle$.

Construction

Stage 0 Define $b_0 = 0$ and $f(0) = a_0$ (where $A = \{a_0, a_1, \ldots\}$) and $n(0) = 0$.

Stage $s+1$ Suppose we've defined $b_0, \ldots, b_{n(s)}$ and $f(b_i) = a_i$ for all $i \leq n(s)$ so that this isomorphism from $\{b_0, \ldots, b_{n(s)}\} = B_s$ to $A_s = \{a_0, \ldots, a_{n(s)}\}$. Furthermore for all i and j, if (b_i, b_j) is a successivity in B_s, then either $|(f(b_i), f(b_j))| = \infty$ (in A) or $(f(b_i), f(b_j))$ is a successivity (in A).

Step 1 Regard B_s as a string by identifying it with its characteristic function. Define a string μ to be *acceptable* if, whenever $\mu(k) = 1$, and $k \in (b_i, b_j)$ (in $\mathbb{Q}$) then $(f(b_i), f(b_j))$ is not a successivity in A (so that $(f(b_i), f(b_j))$ is infinite in A). Find the least string γ, if any extending B_s such that one of the following holds:

(i) There is a z such that for all acceptable τ extending γ, $\Phi_s(\tau;\, z)\uparrow$.

(ii) For some z we have $\Phi_e(\gamma;\, z)\downarrow$ and $\neq \Phi_i(A;\, z)$.

If τ does not exist then $\Phi_i(A)$ would be computably, since to compute $\Phi_i(A;\, z)$ for any z, we need only find some acceptable $\rho > B_s$ such that $\Phi_e(\rho;\, z)\downarrow$ (one must exist). Then we know $\Phi_s(\rho;\, z) = \Phi_i(A;\, z)$. (We will only use acceptable extensions of B_s at each s.)

Our action is to set $B^*_{s+1} = \gamma$ if γ exists, and $B^*_{s+1} = B_s$ otherwise. That is, for each x with $\gamma(x) = 1$ put x into B^*_{s+1}.

Step 2 Now suppose $B^*_{s+1} = \{b_0, \ldots, b_m\}$. For those $j > n(s)$ with $j < m$ we extend f in the obvious way. We put b_j into the correct position in B^*_{s+1} determined by $\mathbb{Q}$ and find a_k in A to map these b_j to. This is possible as the string γ is acceptable. Now for each interval (b_i, b_j) in B^*_{s+1} with $(f(b_i), f(b_j))$ finite (in A) add new elements to make B_{s+1} with the property that if $(b_i, b_j) \in B_{s+1}$ then $|f(b_i), f(b_j)| = 2$ or ∞.

End of Construction

Evidently the construction succeeds since nothing we do at stage $t > s$ injures any action we did for R_s. Thus we diagonalise all the $R\langle e, i\rangle$ and hence $\deg(B) \cap \deg(A) = \mathbf{0}$. Clearly we have made f an isomorphism. $\square$

Actually the argument above holds for quite a wide class of structures and only relies on extensions of embeddings. Richter proved a general structure theoretical result which has the linear ordering result as a particular case.

Definition 3.10 (Richter [188, 189]) We define the *computable embedding condition* as follows. Given a structure A, a finite structure B and an embedding $f : B \mapsto A$, define the class $A_{C,f}$ to be

$$\{D : D \text{ is a finite structure extending } C \text{ embeddable into } A \text{ via a map extending } f\}.$$

Then A satisfies the computable extension property iff for all structures C isomorphic to a finite substructure of A, and for all functions f embedding C into A, the class $A_{C,f}$ is infinite and computable.

Theorem 3.11 (Richter [188, 189]) *For any countable structure A satisfying the computable extension property, there is a countable structure C isomorphic to A such that $degA \cap \deg C = \mathbf{0}$.*

The proof of Theorem 3.11 is similar to that of the Theorem 3.9. Furthermore one can show without too much difficulty that Theorem 3.9 is a consequence of Theorem 3.11.

We remark that some structures can have degrees $\neq \mathbf{0}$, for example groups and lattices (see Richter [188]). We will look at the results there in more detail in Section 9.

The most fundamental question concerning presentations of linear orderings is

Question 3.12 (Open) *Classify the order types ρ such that ρ contains a computably member.*

Of course in Question 3.12 we'd like to have a classical classification. Certain things are obvious. For instance if $\mathbf{A}$ and $\mathbf{B}$ are computably presentable so is $\mathbf{A}+\mathbf{B}$, $\mathbf{A}\cdot\mathbf{B}$, etc. It might appear from the above that we'd need to use n-blocks to make bad isomorphism types. Even here the answer is no.

Definition 3.13 Call a linear ordering $\mathbf{A}$ *discrete* if A contains no limit points.

It follows from a result of Watnick that there are Π_1 discrete linear orderings not isomorphic to computably ones. Watnick's theorem is:

Theorem 3.14 (Watnick [225], Downey [45]) $\mathbf{A}$ *is a* Π_{n+2} *linear ordering iff* ζA *is a* Π_n *linear ordering.*

Watnick's theorem has an interesting history in the sense that it has been rediscovered several times in different contexts (e.g. Downey [45], Ash-Jockusch-Knight [8]). To get a discrete Π_1 linear ordering not isomorphic to a computably one, one takes Watnick's theorem and applies it to a Π_3 linear ordering A not isomorphic to a Π_2 one. Then ζA is a discrete Π_1 linear ordering and not isomorphic to a computably one.

We will look at the proof of Watnick's theorem in Section 5.

Along similar lines, Downey and Knight proved a "1-jump" version of Theorem 3.14.

Definition 3.15 Let A be a linear ordering. Define $\gamma(A) = (\eta + 2 + \eta)A$.

Theorem 3.16 (Downey and Knight [50]) $\mathbf{A}$ *is a* Π_{n+1} *linear ordering iff* $\gamma(A)$ *is a* Π_n *linear ordering. Furthermore* A *is* $\mathbf{a}'$ *presentable iff* $\gamma(A)$ *is* $\mathbf{a}$*-presentable.*

Proof Let $n = 0$ and then relativise. Let A be a Π_1 linear ordering so that A is a Π_1 subset of $\mathbb{Q}$ (identified with $\mathbb{N}$ via a Π_1 subset of $\mathbb{Q}$ using Gödel numbers). Hence $x \in A$ iff $(\forall s)(R(x, s))$. Begin with a copy of $B = (\eta+2+\eta)\eta$. We can identify the x-th member of $\mathbb{Q}$ (i.e. under Gödel numbering) with a canonical successivity in $(\eta+2+\eta)\eta$ is the obvious way. Now perform a construction of "densifying" the ordering B by making the 2 in the x-th position become dense at stage s if $\neg R(x, s)$ holds. So the only remaining points at the end will correspond to the $x \in A$. □

Corollary 3.17 *There exists a* Π_1 *linear ordering* A *not isomorphic to a computably one with and* A *having order type* $(\eta+2+\eta)\tau$ *for some* τ*. In particular* A *has no* n*-blocks for* $n \neq 2, 1$ *and no* ω *or* ω^* *intervals.*

We remark that Downey and Moses [53] have proved that every low discrete linear ordering is isomorphic to a computable one.

One quite natural approach is to look at increased levels of effectivity. An important notion is that of a *decidable* structure. This is one whose whole diagram is computable. That is, given any sentence we can effectively compute its validity in the structure. Along similar lines one can define a structure to be n-computable if we can decide effectively an n quantifier sentences. It is not hard to construct a computable ordering that is not 1-computable. We warm up with such a result.

Theorem 3.18 (Folklore) *There is a computable linear ordering A of order type ω with $S(x)$, the successor function, not computable. Hence A is not 1-computable since the adjacency relation is not computable.*

Proof We give this result as a simple introduction to the priority method, and for this reason we don't give the simplest proof.

We shall build our computable order $A = \cup_s A_s$ in stages. We label the points $\{a_0, a_1, \ldots\}$ for convenience. To make A have order type ω it suffices to satisfy, for each $e \in \mathbb{N}$, the requirement

$$N_e : \ a_e \text{ has } < \infty \text{ many predecessors.}$$

To make sure that $S(x)$ is not computable in A, we diagonalise over all possible algorithms for $S(x)$. To do this we satisfy for each $e \in \mathbb{N}$

$$P_e : \ \varphi_e \text{ is not the successor function on } A.$$

Here $\{\varphi_e\}_{e \in \omega}$ lists all p.r. functions. To meet P_e the basic idea is this. We pick a *follower* $x = x(e)$ and wait for a stage s where $\varphi_{e,s}(x)\downarrow$. If this never happens, then we win as φ_e is not total. If $\varphi_{e,s}(x)\downarrow$ we make sure that $\varphi_e(x)$ is not the successor of x by placing, if necessary, a new point $(s+1)$ between x and $\varphi_e(x)$.

This action gives potential conflict with N_j since if $\varphi_e(x) = a_j$ it may be that we put a new point to the left of a_j. If infinitely many P_e *together* were allowed to do this to a *single* a_j then we would fail to satisfy $\mathbb{N}_j :$ a_j would have infinitely many predecessors.

The idea then is to have a priority ranking

$$N_0, \ P_0, \ N_1, \ P_1, \ \ldots$$

of the requirements. We then only allow P_i to "injure" N_j if $i < j$. Here 'injury' means 'add a point to the left of a_j'. If we ensure that the total action we perform for each P_i within $i < j$ only injures N_j finitely often, then N_j will be happy since after a certain stage s_0 no new predecessors of a_j will be added. Note that we can't predict s_0 but when this *finite injury priority method* is used one argues by simultaneous induction that all the P_j 'require attention' only finitely often and therefore all the N_e are injured only finitely often and are (hence) met.

The device that ensures we can do this is called the *coherence strategy* and the action we perform for a single requirement is called the *basic module*. We have described the basic module above. The coherence strategy here is very simple: we only pick followers x of P_k $(k \geq j)$ is x is not a predecessor of a_j. Then P_k will never add a predecessor to a_j (if $k \geq j$).

3.19 The formal construction looks as follows. We say P_e *requires attention* at stage $s+1$ if e is the least such that one of the following options holds.

3.20 P_e has no follower.

3.21 P_e has a follower x and $\varphi_{e,s}(x)\downarrow$ and $\varphi_{e,s}(x)$ is the successor of x.

Construction

Stage 0 Set $A_0 = \{a_0\}$.

Stage $s+1$ We are given A_s as $a_0, \ldots, a_s$ in some specified order. Find the $e \leq s$ with P_e requiring attention (there will be one).

If (3.21) pertains it will be via some $x = a_{i_s}$ with $\varphi_{e,s}(x) = S(x) = a_{i+1_s}$ for some a_{i+1_s}. Put a_{s+1} between a_{i_s} and a_{i+1_s} (killing $\varphi_{e,s}$ forever). Go to stage $s+2$.

If (3.20) pertains, put a_{s+1} to the right of A_s (i.e. $a_i < a_{s+1}$ for all $i \leq s$), and declare a_{s+1} to be a follower for P_e.

End of construction

It is not difficult to see that each P_e requires attention at most twice, and all the P_e and N_e are met so that A has the desired properties. $\square$

We spent a long time with the easy result above in the hope that the reader will understand the more elaborate uses of the machinery (and its extensions) later. In the future we will only sketch such proofs by specifying the basic modules and the coherence strategies.

Sometimes the concepts of n-computability and decidability are more closely related than the obvious fact that A is decidable iff it is uniformly n-computable for all $n \in \mathbb{N}$. For instance, the old result of Langford [126], shows the following. Recall that L is called discrete iff $L \cong$ $eta.\widehat{L}$ for some $\widehat{L}$.

Theorem 3.22 (Langford [126]) *A discrete linear ordering is decidable iff it is* $1-$ *computable.*

The proof of Langford's Theorem uses quantifier elimination. We illustrate this technique in the next result which completely classifies $1-$computability for computable linear orderings.

Theorem 3.23 (Moses [157, 158]) *A computable linear ordering is* $1-$*computable iff the collection of adjacencies of L is a computable set.*

Proof Sketch. First one shows that if φ is a quantifier free sentence on variables $x_1, ..., x_n$ then φ is equivalent to a disjunction of formulae of the form $\nu_1 \wedge ... \wedge \nu_n$ where each ν_i is of the form $x_{\pi(i)} < x_{\pi(i+1)}$ or $x_{\pi(i)} = x_{\pi(i+1)}$ for some permutation π of $\{1, ..., n\}$. We call this the standard form of φ. It follows that an existential formulae γ is equivalent to a disjunction of the form

$$\exists \overline{x} \psi_1(\overline{x}, \overline{y}), \vee ... \vee \exists \overline{x} \psi_m(\overline{x}, \overline{y}),$$

where each ψ_i is in standard form. Thus to se if γ is satisfiable, we need only look at each $\exists \overline{x} \psi_i(\overline{x}, \overline{y}) = \gamma_i$, say.

It is easy to see that each γ_i is equivalent to a finite set of statements of the form "$y_i = y_j$", $y_i < y_j$, "there are at least $k(\leq n)$ elements between y_i and y_j," "there are at least k elements

$> y_i$," and "there are at least k elements $< y_i$." These are decidable iff the collection of succesivities is computable. $\square$

Moses [unpubl.] has a classification hierarchy akin to the above for general orderings. We remark that Moses [161] has constructed for each n an n-computable not isomorphic to a $n+1$-computable one. ordering Chisholm and Moses have also proven the following.

Theorem 3.24 (Chisholm and Moses [31]) *There exists a linear ordering $\widehat{L}$ which is n-computable for each $n \in \mathbb{N}$ but $\widehat{L}$ not isomorphic to a decidable linear ordering.*

Here are some related questions.

Question 3.25 *(i) Are there n computable boolean algebras not isomorphic to $n+1$ computable ones for each $n \in \mathbb{N}$? What about a boolean algebra which is n computable for each $n \in \mathbb{N}$ but is not isomorphic to a decidable one?*
(ii) For each $n \in \mathbb{N}$, is there a finitely presented group which is n-computable but not $n+1$ computable? Can we construct a nondecidable finitely presented group which is n-computable for each $n \in \mathbb{N}$?

Proof Sketch of Theorem 3.24. We sketch the proof of Theorem 3.24 particularly since it illustrates Ehrenfeucht-Fraisse games. The argument is a finite injury priority argument. For structures P and Q we will write $P \equiv_{\Sigma_n^0} Q$ to mean that P and Q satisfy the same Σ_n^0 sentences. The idea is to build a sequence

$$L_0, L_1, L_2, ...$$

of decidable linear orderings with limit $\widehat{L}$ so that for each n and each $m > n$, we ensure that for all $\overline{a} \in \widehat{L}$ we can effectively find a $\overline{b} \in L_m$ such that $\langle \widehat{L}, \overline{a} \rangle \equiv_{\Sigma_n^0} \langle L_m, \overline{b} \rangle$ and conversely. This will ensure that $\widehat{L}$ is n-computable by the decidability of L_n and playing an Ehrenfeucht-Fraisse game between $\widehat{L}$ and L_n.

To ensure that $\widehat{L}$ has no decidable copy, we will ensure that L_n is one of L_n^+ or L_n^-. (As usual, we are building all the orderings in stages and at each stage they are finite.) We begin by building $L_n = L_m^-$ and only change when we see an opportunity to diagonalize. We will have a specific sentence φ_n used to distinguish between L_n^+ and L_n^-. $L_n^+ \models \varphi_n$ and $L_n^- \not\models \varphi_n$. We ensure that the theory of $\widehat{L}$ is not computable and hence $\widehat{L}$ cannot have a computable copy. Thus the process and the switching is simply controlled by an opponent trying to enumerate the theory of $\widehat{L}$ and we wait till the opponent enumerates $\neg\varphi_n$ to switch.

Let T_1,...,T_k be orderings. Then

$$\sigma(T_1, ..., T_k)$$

denotes the *shuffle sum* of $T_1, ..., T_k$ which is defined by taking a copy of $\mathbb{Q}$ and densely putting copies of each of the T_i's in. (So between T_i and T_j there is always a copy of T_q for any i, j, q.)

Now $L_0 = \sigma(1, 2, 3)$. For sequence $1, ..., k$, let $1, ..., \widehat{j}, ..., k$ denote the subsequence obtained by leaving out j. We define $L_1^0 = \sigma(1, 2, 3)$, and define $L_1^1 = \sigma(2, 3)$, $L_1^2 = \sigma(1, 3)$, and $L_1^3 = \sigma(1, 2)$. Finally define

$$L_1^- = \sigma(L_1^0, L_1^1, L_1^2) \text{ and } L_1^+ = \sigma(L_1^0, L_1^1, L_1^2, L_1^3).$$

More generally, we will know that

$$L_{n-1} = \sigma(T_1, ..., T_k).$$

We will let $L_n^0 = L_{n-1}$, and

$$L_n^j = \sigma(T_1, ..., \widehat{T}_j, ..., T_k).$$

Then, as above,

$$L_n^+ = \sigma(L_n^0, ..., L_n^{k-1}) \text{ and } L_n^- = \sigma(L_n^0, ...L_n^k).$$

Notice that in a stage by stage construction we can switch from L_n^- to L_n^+ easily. Also note that each L_n^j misses an L_{n-1}^p for some p. (This is what the T_i's are.) However, each L_{n+1}^w has all the $L_{n_1}^p$'s appearing in arbitary order. One now proves by induction that each L_n^j with $n > 1$ satisfies a certain Π_{n+1} sentence φ_n^j that is not satisfied by any L_n^k for $k \neq j$ nor any shuffle of them. (For instance, φ_1^j says that there is no maximal block of size j.) Second one proves by induction that all the L_n^p are $\equiv_{\Sigma_{n-1}^0}$ to each other or any shuffle of each other. The result will then follow by the diagonalization of the construction, and these two facts. □

Corollary 3.26 (Chisholm and Moses [31]) *There is a structure that is intrinsically n-computable for each $n \in \mathbb{N}$ (i.e. each computable structure isomorphic to it is n-computable) yet is not isomorphic to a decidable structure.*

Proof. Take the ordering $\widehat{L}$ above and add constants c_i for each member of the ordering. □

4 Computable Isomorphisms

A natural suggestion would be to look at 'effective' order types instead of classical order types. That is, since we are, after all, looking at effective procedures on computable orderings, surely the 'correct' classification would be to consider those computable linear orderings computably isomorphic to a given one. Does this make a difference? As we shall see in this (and later) sections, it certainly does. We have already seen that computable properties are not, in general, invariant. For instance, the natural presentation of $\omega = \{0 < 1 < \cdots\}$ has the property that the successor function $S(x)$ is computable. Heowver in Theorem 3.18 we observed that there is a computable copy of ω with $S(x)$ not computable. It follows that there are two isomorphic computable linear orderings that are not *computably isomorphic* (as $S(x)$ being (non) computable would be preserved by a computable isomorphism). Consideration of this example leads to the following basic definition.

Definition 4.1 We call a computable linear ordering A *computably categorical* (sometimes *autostable* in the literature) if any computable linear ordering B isomorphic to A ($B \cong A$) is computably isomorphic to A ($B \cong_{\text{comp}} A$).

Clearly all finite linear orderings are computably categorical. Some infinite ones are too.

Example 4.2 (Folklore) *η is computably categorical.*

To see this we effectivise Cantor's back-and-forth argument. Let $A = \{a_0, a_1, \ldots\}$, $B = \{b_0, b_1, \ldots\}$ be two dense computable orderings without end points. We define $f_0(a_0) = b_0$. At stage $2s$ we ensure that $a_s \in \mathrm{dom} f$. To do this we will have defined $\mathrm{dom} f_s = a_0, \ldots, a_{s-1},, a_{i_0}, \ldots, a_{i_{s-1}}$ in some order with f_s a partial isomorphism. Now assuming $a_s \notin \mathrm{dom} f_s$, we see where a_s lies relative to the points in $\mathrm{dom} f_s$. If, for instance, $a_i < a_s < a_j$ we simply use the density of B to find some b_k with $f_s(a_i) < b_k < f_s(a_j)$ and define $f_{s+1}(a_s) = b_k$. At odd stages $2s + 1$ we similarly ensure that $b_s \in raf$ by using the density of A. By extensions of embeddings, $f : A \cong_{\text{comp}} B$. □

The computably categorical linear orderings were classified by Remmel [182]:

Theorem 4.3 (Remmel [182]) *A computable linear ordering* $\mathbf{A}$ *is computably categorical iff the (classical) order type of* $\mathbf{A}$ *has only a finite number of successivities.*

Proof If $\mathbf{A}$ is a computable linear ordering with only finitely many successivities then $\mathbf{A}$ has order type $\left(\sum_{i=1}^{q} n_i + \eta\right) + n_{q+1}$ with $n_i < \infty$ for all i. To show that $\mathbf{A}$ is computably categorical, suppose $\mathbf{B} \cong \mathbf{A}$ with $\mathbf{B}$ computable. then $\mathbf{B}$ must also have the same type and hence if we match the n_i in $\mathbf{B}$ to those in $\mathbf{A}$ we can extend to a computable isomorphism by the technique of Example 4.2.

Conversely, suppose $\mathbf{A}$ has infinitely many successivities. We build $\mathbf{B} \cong \mathbf{A}$ with $\mathbf{B} \not\cong_{\text{comp}} \mathbf{A}$. To do this, we use a priority argument and meet the requirements ($e \in \mathbb{N}$):

$$N : \mathbf{B} \cong \mathbf{A}$$

$$P_e : \ \varphi_e \text{ is not an isomorphism from } \mathbf{B} \text{ to } \mathbf{A}.$$

We will build $\mathbf{B}$ in stages as $\mathbf{B} = \cup_s B_s$ and will ensure that $f : \ \mathbf{B} \to \mathbf{A}$ is a (Δ_2-) isomorphism by ensuring that we meet (for all $e \in \mathbb{N}$):

$$N_e : \ \lim_s f_s(b_e) \text{ exists}$$

(i.e. we have $f_{s+1}(b_e) \neq f_s(b_e)$ only finitely often),

$$M_e : \ f(b_e)\text{is defined (i.e. } f \text{ is a function)}$$

$$\widehat{M_e} : \ (\exists y)(f(y) = a_e) \text{ (i.e. } f \text{ is onto).}$$

Additionally we ensure that f in injective. The key to the construction is the basic model for P_e.

The idea is as follows. If we ever see φ_e not 1-1 (or if for some x, $\varphi_e(x_i) \uparrow$ although we can't know this) we need not worry about φ_e. Assuming these don't happen, we really need to do something to meet P_e. At stage s we will have ensured that f_s is a partial isomorphism taking $B_s = \{b_0, \ldots, b_s, b_{i_0}, \ldots, b_{i_s}\}$ to $\{a_0, \ldots, a_s, a_{i_0}, \ldots, a_{i_s}\} = A_s$. (This meets M_e and $\widehat{M_e}$ provided we meet N_e.)

Suppose in A_s we *knew* that (a_i, a_j) was a successivity. Let $f_s(b_k) = a_i$ and $f_s(b_m) = a_j$ say. For the sake of P_e we need do nothing until we see a stage $t > s$ such that for some b_n, b_r we have

4.4 (i) $\varphi_e^t(b_n) \downarrow$ and $\varphi_e^t(b_r) \downarrow$

(ii) $\varphi_e(b_n) = a_i$ and $\varphi_e(b_r) = a_j$

(iii) (b_n, b_r) is a successivity in B_t.

Now we can force φ_e to be wrong at stage $t+1$ by putting a new element b_g between b_n and b_r, since then (b_n, b_r) is not a successivity in B yet (a_i, a_j) is a successivity in A.

Obviously, if we are to keep $f = \lim_s f_s$ an isomorphism, the strategy above will entail us redefining f_{t+1}. For instance, we may have $f_{t+1}(b_n) = a_u$ and $f_{t+1}(b_r) = a_v$. When we put b_g between b_n and b_r unless a new element appears between a_u and a_v, we'd need either to define $f_{t+1}(b_g) = a_u$ or $f_{t+1}(b_g) = a_v$ and hence either make $f_{t+1}(b_n) \neq f_t(b_n)$ or $f_{t+1}(b_r) \neq f_t(b_r)$ (as we need f injective). Indeed, it is possibly the case that we will need to reset $f_{t+1}(b)$ either for all $b \in B_{t+1}$ with $b \geq b_r$ or for all $b \in B_{t+1}$ with $b \leq b_n$.

Here is where we use a priority argument. We do not allow this action for P_e to take place if it injures N_j for $j < e$. That is, we don't let P_e make $f_{t+1}(b_j) \neq f_t(b_j)$ for any $j < e$. Now this causes the following problems. Suppose that we have points b_i and b_j both with $i, j < e$ and it happens that $b_i \leq b_n < b_r \leq b_j$. Now it may be that we have defined both $f_t(b_i)$ and $f_t(b_j)$. Since $i, j < e$ they have higher priority than P_e and so *we cannot add a new point* b_g *between* $f_t(b_i)$ *and* $f_t(b_j)$ *at any stage* $\geq t$. Of course, *if such a new point appears at some stage* $t_1 > t$ we are free to pursue the strategy of the basic module and still keep $f_{t_1+1}(b_i) = f_t(b_i)$ and $f_{t_1+1}(b_j) = f_t(b_j)$. That is, *only change* f_{t_1} *on the open interval* (b_i, b_j).

The trouble is that we *cannot know* if another point will appear, but we now have an obvious strategy: while we *wait* for some such new point to appear we begin a *new strategy on a new* successivity in A. If the interval $f(b_i),\ f(b_j)$ is finite, after a finite number of such tries we will get a pair of points $(b_{i_1},\ b_{i_2})$ *outside* of the *finite* interval $(b_i,\ b_j)$. If the interval is *infinite*, then eventually we will see a new point appear and can *return* to the old strategy. In this way we can make the N_j and P_e strategies cohere since we will get to meet P_e on *some* pair.

The final problem not addressed in the above is the following: we may not be able to computably identify the successivities in A. Being a successivity is only Π_1 and it is not clear that A may be isomorphic to a computable linear ordering with computable successivities. Indeed this is not always the case:

Theorem 4.5 (Remmel, Goncharov, after Feiner) *There is a computable linear ordering A such that if B is computable and $B \cong A$ then B does not have computable successivities.*

Proof (Downey) One way to get Theorem 4.5 is to use the following. Take a Π_1 linear ordering C not isomorphic to a computable one. Now apply the Downey-Knight [50] construction 3.15 to get $A = (\eta + 2 + \eta)C$ computably presentable. Then the order type of successivities of A are isomorphic to C and if they were computably presentable so would C be, which it is not. □

Proof of Theorem 4.3 concluded To overcome this final problem, we *guess* if $(a_i,\ a_j)$ is a successivity. If it is not, as we will find out, we abandon it and try another pair $(a_{i_1},\ a_{j_1})$. The trick is to list all the pairs $(a_i,\ a_j)$ by Gödel numbers, and let P_e always use the *least* pair. Clearly we will eventually settle on a real successivity. To complete the proof one simply formalises the above, and the result will follow by a standard application of the finite injury technique. This concludes our sketch of the proof of 4.3. □

It is not difficult to interweave other 'finitary' requirements with the above argument. For example, if $L_1,\ L_2,\ \ldots$ is any computable list of isomorphic linear orderings one can diagonalise against each of them simultaneously.

Corollary 4.6 (Remmel [182]) *(i) A classical order type of a computable linear ordering contains either one, or infinitely many computable order types.*

(ii) If L is a computable linear ordering with infinitely many successivities then there is no computable list $L_1,\ L_2,\ \ldots$ of all computable linear orderings isomorphic to L.

Corollary 4.6 seems an appropriate place to mention another recurrent theme in computable combinatorics: *the effective content of classical theorems.* We will look at this theme in more detail in the sections to follow, but will confine ourselves here to some brief comments on Corollary 4.6.

A famous and deep combinatorial theorem is Lavev's theorem [128] on the better-quasi-ordering of countable linear orderings under embeddability. One form of this result is:

Theorem 4.7 (Lavev [128]) *Let $\rho_1,\ \rho_2,\ \ldots$ be a countable collection of countable order types. Then there is some $i < j$ with ρ_i embeddable into ρ_j.*

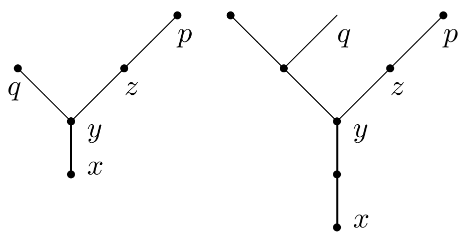
Corollary 4.6 says that the obvious computable analogue of Theorem 4.7 fails for computable order type. That is, if $\rho_1,\ \rho_2,\ \ldots$ are computable order types they may all be pairwise *computably* nonembeddable. (Although 1 only speaks of nonisomorphism, the proof of Theorem 4.3 actually gives computable nonembeddability.)

Since we know some ρ_1 embeds into ρ_j one might ask exactly how complicated such an embedding would be. For example, must there be an arithmetical embedding? Richard Shore [**?**] has the decisive material here. He has shown that it is possible to choose the collection of order types so that there is not even a hyperarithmetical witnessing function.

At this point I cannot help but mention some beautiful recent developments related to Theorem 4.7. One of the great achievements of the twentieth century is Gödel's famous incompleteness theorem which roughly states that in any sufficiently strong system with an computably enumerable set of axioms, there is a formula statable *within* the system, true of the system, but not provable *within* the system. Unfortunately, whilst Gödel's theorem has a major impact on the foundations of mathematics, many 'working' mathematicians regarded it as irrelevant to their studies, since the formulae found by Gödel were 'not interesting' and were 'things one would not wish to prove'.

First Paris and Harrington [171] and then Harvey Friedman and others (see Simpson [205]) destroyed this thesis by producing 'unprovable "mathematical" theorems' in a number of systems. We will confine ourselves to a single example.

Let T_1 and T_2 be finite trees regarded as lattices. We say T_1 embeds into T_2 if there is an injective function $f : T_1 \to T_2$ preserving meets (i.e. $f(x \wedge y) = f(x) \wedge f(y)$). See the example in the diagram below:



T_1 T_2

Kruskal[114] proved that if T_1, T_2, ... is an infinite collection of finite trees, then there is some $i < j$ with T_1 embeddable into T_j. This statement is not provable in 'finite combinatorics' (i.e. Peano arithmetic (PA)) but neither is it *statable* in PA. Friedman found the following finite form of Kruskal's theorem:

Theorem 4.8 (H. Friedman) *Let n be given. There exists a number $k = f(n)$ so large that if $T_1, \ldots, T_k$ are finite trees with $|T_i| \leq n \cdot i$, then there exists $i < j$ with T_i embeddable into T_j.*

Whilst Theorem 4.8 is clearly statable in PA, Friedman proved that 4.8 cannot be *proven* in PA and any proof of 4.8 necessarily involves *infinite* sets. (Indeed Friedman found variations of 4.8 that need *uncountable* sets.) The way to prove this is to show that the witness function $n \to f(n)$ grows so fast as to not be provable in PA. The reader is referred to Simpson [205] and Frieman-Simpson-Smith[71] for further results. These logical investigations have had some fine consequences in classical combinatorics. For example, Friedman's result above was a crucial ingredient of the Robinson-Seymour theorem that the set of all finite graphs is well-quasi-ordered under the relations 'H is minor of G', and as a consequence there is a 'Kuratowski' theorem for surfaces.

Returning to our story, Remmel's theorem 4.6 has been generalised in several ways. First we could ask the more general question of classifying when a computable linear ordering is Δ_n-*categorical* for some n. That is, try to classify (by order type ρ) when, given any two computable linear orderings A and B of type ρ there is a Δ_n-isomorphism from A to B. Similarly we might ask when *every* isomorphism is Δ_n. This latter property is called Δ_n-*stability.* For instance it is easy to see that ω *is* Δ_2-*stable and (hence)* Δ_2-*categorical.*

This leads to the powerful recent techniques developed by Chris Ash and by others. We will look at these in Section 9, when we will be able to sketch the technical machinery needed for Ash's results.

A second method of generalisation is to look at linear orderings as merely an example of a more general class of structures. This approach was taken by Ash-Nerode [11], Goncharov [78], Barker [14], Harizanov [83, 84, 85] and others using *Model Theory.* Since we are trying to keep this account relatively 'nonlogical' again we give only a bare outline. A model is a structure $(A, R_1, \ldots, R_n, \ldots, F_1, \ldots, F_n, \ldots, C_1, \ldots, C_n, \ldots)$ where A is the universe, the R_i are relations on A, the F_i are functions of A and the C_i are distinguished constants. For example, a group is structure of the form $G = (G, \cdot, 1)$. In the language of the structure we can write sentences and study all of the structures obeying those sentences. The class of models of the sentences

$$(\forall x,\ y,\ a)\ (x \cdot (y \cdot z) = (x \cdot y) \cdot z)$$

$$(\forall x,\ y)\ (x \cdot y = y \cdot x)$$

$$(\forall x)\ (\exists y)\ (x \cdot y = y \cdot x = 1)$$

$$(\forall x)\ (x \cdot 1 = x)$$

is simply the set of abelian groups and a model of these sentences is an Abelian group. A model is computable if $A = \mathbb{N}$ and the R_i, F_i and C_i are a computable collection of computable relations, functions and constants respectively. Dzgoev and Goncharov [56] discovered a condition called *branching* and showed that if A is a computable model which branches, then Q is not computably categorical. This result has applications including Remmel's theorem and similar results classifying the computably categorical boolean algebras (La Roche, Goncharov) and p-groups (Smith). The only problem with Dzgoev and Goncharov's (and other) results is that it is often quite difficult to verify that a given structure branches. (Sometimes this is much more difficult than a direct proof.) One open question here is:

Question 4.9 (Open) *Develop a generalisation of branching for* Δ_n*-categoricity.*

A very attractive result here is due to Moses. If R is a relation on a computable linear ordering $\mathbf{A}$, we call R *intrinsically computable* if in every computable linear ordering $\mathbf{B}$ classically isomorphic to $\mathbf{A}$ via $F: A \cong B$ we have $f(R)$ computable. (The notion is due to Ash-Nerode [11].)

Theorem 4.10 (Moses [157]) *If* $\mathbf{A}$ *is a computable linear ordering then a computable relation* R *on* $\mathbf{A}$ *is intrinsically computable iff* R *is equivalent in* $\mathbf{A}$*to a quantifier free formula* $\theta(x_1 \ldots x_n,\ a_1 \ldots a_n)$ *with* $a_1, \ldots, a_n$ *in* $\mathbf{A}$.

The final generalisation of Remmel's theorem we will look at is to study the 'fine structure' of isomorphism types of computable orderings and see what can be preserved. In the remaining sections we see several of these types of results. For this sections we will only look at the situation where we equip the linear orderings with additional computable properties. First we know that not all linear orderings have copies with computable successivities. Remmel [177, 180, 181] has a number of results in this area as corollaries to his results on boolean algebras. These investigations are related since as we mentioned earlier any computable boolean algebra can be represented as the boolean algebra of the left closed right open subintervals of a computable linear ordering (similarly Δ_n, Σ_n, Π_n). Here is a quick proof.

Theorem 4.11 (The Representation Theorem, folklore after Stone) *Every countable boolean algebra is isomorphic to a interval algebra $Intalg(L)$ of a linear ordering of the same degree.*

Proof. This is just an effective version of Stone's theorem. We represent a given B as an algebra of sets. Thus let $B = \cup_s B_s$ with $B_0 = \{0, 1\}$, and $B_{s+1} - B_s = \{b_s\}$. We will define at each stage a subalgebra $\widehat{B_s}$ containing B_s.

Define L_0 the ordering with two points labelled 0 and 1. (Then $Intalg(L_0)$ consists of the two sets $\emptyset$ and $[0, 1)$.) So we have the induced mapping g_0 with $0 \mapsto \emptyset$ and $1 \mapsto [0, 1)$. At stage $s + 1$ we will have a set of $Atoms_s = \{a_{s_1}, ..., a_{s_n}\}$ listing the atoms of the subalgebra of $\widehat{B_s}$ together with the linear ordering $L_s = 0, x_{s_1}, ..., x_{s_n} = 0$ so that $g_s(a_{s_j}) \mapsto [x_{s_{j-1}}, x_{s_j})$ inducing an isomorphism from the subalgebras $\widehat{B_s}$ to $Intalg(L_s)$.

At stage $s + 1$ if b_s is in $\widehat{B_s}$ we need to do nothing. Otherwise, for each $a = a_{s_i}$ such that b_s *splits* a (i.e. both $x \wedge b_s$ and $x \wedge \overline{b_s}$ are nontrivial), add a new point y to L_{s+1} between $x_{s_{j-1}}$ and x_{s_j} to split the interval $[x_{s_{j-1}}, x_{s_j})$ into $[x_{s_{j-1}}, y) \cup [y, x_{s_j})$. Naturally we map $a_{s_j} \wedge b_s$ to one of them, say, $[x_{s_{j-1}}, y)$ and map $a_{s_j} \wedge \overline{b_s}$ to $[y, x_{s_j})$. Note that this generates two new atoms for $\widehat{B_{s+1}}$. And of $Intalg(L_{s+1})$ Let $c_1, ..., c_m$ denotes the atoms of $\widehat{B_{s+1}}$ below b_s. Clearly we have ensured that the induced map $g(b_s) = g(c_1) \cup ... \cup g(c_m)$ works. The results follows. $\square$

Notice that in the above atoms correspond to successivities, so when Remmel studied computable boolean algebras with computable atoms he had corollaries about computable linear orderings with computable successivities. Unfortunately there is no direct transfer the other way since many nonisomorphic linear orderings give rise to the same boolean algebra.

Question 4.12 (Open) *Determine the relationship between the degrees of linear orderings and the boolean algebras they generate.*

If a linear ordering has computable successivities, then we can sharpen Corollary 4.6.

Theorem 4.13 (Remmel [182]) *If A is a computable linear ordering with an infinite computable set of successivities, then*

(i) There is a computable collection B_1, B_2, ... of computable linear orderings each isomorphic to A such that if $i \neq j$ then the sets of successivities of B_i and B_j are Turing incomparable.

(ii) *We can arrange that the sets of successivities S_i of each B_i are immune (that is, there is no infinite computably enumerable subset of S_i).*

As we will see in Section 7, Theorem 4.13 1 does *not* hold if we drop the 'computable successivities' hypothesis. The proofs of 1 and 2 (and similar results) come from finite injury arguments blended with the techniques used in Corollary 4.6. See Remmel [182] for details.

Another variation comes from restricting the sorts of structures we look at. A nice question here was asked by Nerode, Remmel and others: classify the order types of computably categorical linear orderings with computable successivities. That is, we want order types ρ such at if A and B have types ρ *and both have computable successivities* then $A \cong_{\text{comp}} B$. This question proved quite hard and was finally solved by Moses [157, 158].

Theorem 4.14 (Moses [157, 158]) *The computably categorical linear orderings with computable successivities (in the sense above) are precisely those with order types* $\sum_{i=1}^{n}(k_i+g_i)+k_{n+1}$ *where* k_i *is finite and* $g_i \in \{\omega, \omega^*, \zeta\} \cup \{d \cdot \eta : d \in \mathbb{N}\}$ *for all* i.

Moses' proof of Theorem 4.14 is not obvious and involves a surprising intermediate step. Recall from the last section that a linear ordering is called *n-computable* if the Σ_n formulae uniformly denote computable relations. Note that '(x, y) is not a successivity' is an example of a Σ_1 relation so to say A has computable successivities is to assert that a *particular* Σ_1-relation on A is computable. Hewever, we recall from section 3 Theorem **??**

A computable linear ordering **A** *is 1-computable iff* **A** *has computable successivities.*

Define a linear ordering to be *good* if it has the order type described in Theorem 4.14. We need the following definition.

Say that **B** is *weakly good* if **B** has order type $\left(\sum_{i=1}^{n} k_i + g_i\right) + k_{n+1}$ with each k_i finite and $g_i \in F_n$ for some $n = n(i)$, with F_n denoting the class of linear orderings with no block of size $> n$.

Definition 4.15 (i) *The* block relation $B(x, y)$ *is defined to hold if there are only finitely many elements between* x *and* y.

(ii) *The relation* $M_k(x)$ *holds if the block containing* x *has exactly* k *elements.*

Moses proved Theorem 4.14 using Theorem 3.23 and the following sequence of results.

Theorem 4.16 (Moses [157, 158]) (i) *If* **A** *is a 1-computable linear ordering with the block relation computable then the order type of* Z *is weakly good iff every 1-computable linear ordering isomorphic to* A *has the block relation computable.*

(ii) *If* **A** *is a 1-computable linear ordering with* $M_i(x)$ *for* $i \in \mathbb{N}$ *all computable then* A *has order type* $\left(\sum_{i=1}^{n} k_i + g_i\right) + k_{n+1}$ *with each* k_i *finite and* $g_i \in \{q \cdot \eta : q \in \mathbb{N}\}$ *iff every 1-computable linear ordering has* $M_i(x)$ *for* $i \in \mathbb{N}$ *computable.*

(iii) *If* **A** *is a computably categorical 1-computable linear ordering, then it has* $B(x, y)$ *computable.*

(iv) *If* **A** *is a computable linear ordering with no blocks of infinite length and* $B(x, y)$ *computable, then the isomorphism type of* **A** *contains a 1-computable linear ordering with* $B(x, y)$*, the predecessor relation* $P(x)$ *computable and the successor relation* $S(x)$ *computable.*

To finish the proof Moses assembles the facts above. First suppose that **A** is of good order type. Then a standard back and forth argument like that used in Corollary 4.6 shows that **A** is computably categorical for 1-computable linear orderings, then by Theorem 4.16(3), **A** has computable block relation and hence **A** is weakly good by (1). We can then partition **A** by a finite number of points $P_0 < \cdots < P_m$ with each interval $(-\infty, P_0)$, (P_0, P_1), $\cdots$, (P_m, ∞) of order type in $F_{n(i)} \cup \{\omega, \omega^*, \zeta\}$.

Let **B** be the result of deleting from **A** all orderings of type ω, ω^*, or ζ. Then **B** is computably categorical, 1-computable and has by 4.16(4) $B(x, y)$, $S(x)$ and $P(x)$ all computable. As **B** has no infinite blocks. This makes $M_i(x)$ computable for all i and hence by 4.16(2) **B** has order type $(\sum_{i=1}^{n} k_i + g_i) + k_{n+1}$ with k_i finite and $g_i \in \{q \cdot \eta : q \in \mathbb{N}\}$. It follows that the order type of **A** is good, and we are done. □

We should note that the proof of Moses' theorem is greatly complicated by the analogous result that complicated Remmel's theorem 4.6. Recall that this was that there are computable linear orderings **A** with no recursive linear orderings in the order type of **A** having computable successivities. Here Remmel [182] used a coding argument (i.e. like Feiner's argument) to construct a 1-computable linear ordering **A** with the property that if **B** is a computable linear ordering isomorphic to **A** then the block relation is not computable in **B**. Moses and Hingston (see Moses [157]) used another coding argument to construct a 1-computable linear ordering not isomorphic to a 2-computable linear ordering. The above result classifies the computably categorical 1-computable linear orderings. Naturally the following question suggests itself.

Question 4.17 (Open) *Classify the computably categorical* n*-computable linear orderings (i.e. amongst* n*-computable linear orderings).*

Schwartz [202] investigated the computable categoricity amongst linear orderings with the block relation computable. He used a finite injury priority argument to show

Theorem 4.18 (Schwartz [202]) *The linear orderings with the block relation computable that are computably categorical in this class are exactly those of order type*

$$f_0 + k_0 + f_1 + k_1 + \cdots + f_n + k_n + f_{n+1}$$

where each f_i *is finite and* $k_i \in \cup_{q \in \mathbb{N}} q\eta$*.*

Proof We follow Schwartz's thesis [202]. (Actually the proof in Schwartz's thesis has a slight flaw which is not difficult to repair, but Schwartz has not published this.) A routine back and forth argument shows that the above order types are sufficient. Conversely, suppose **A** is an order type not of the above form. Then we must build $B \cong A$ such that B has a computable block relation and $B \not\cong_{\rm rec} A$. So we must ensure that

$$R_e : \varphi_e(B) \neq A \text{ if } \varphi_e \text{ is the } e\text{-th computable partial isomorphism.}$$

There are two cases:

Case 1 For some $a \in A$, the block of a is infinite. So a lies in a block of type ω, ω^* or $\omega^* + \omega$. In this case it is not difficult to see that we can build $B \cong A$ by using the identity outside of a's block and the Remmel strategy on a's block to kill all the φ_e. Note that the block relation is computable as a's is.

Case 2 Otherwise. Each block in A is finite. It is not hard to see that as A is infinite the order type of A is $\Sigma \{h(q) : q \in \mathbb{Q}\} + h(+\infty)$ where h is a function from $\mathbb{Q} \cup \{-\infty, \infty\}$ to $\mathbb{N}$.

The strategy is to diagonalise against all the φ_e. The basic idea is to play φ_e on blocks. That is, for two points x and y in B ensure:

$$|[x, y]| \text{ (in } B) \neq |[\varphi_e(x), \varphi_e(y)]| \text{(in } A).$$

There are two subcases

Case 2a

ra$h$ is unbounded. The strategy here is quite simple. In A wait till we see two points a and b in the same block and wait till $\varphi_e(x) = a$ and $\varphi_e(y) = b$ for some x and y. Now we can use a Remmel strategy. Wait until we see some block $[a_1, b_1]$ occur in A with more that $|[a, b]|$ many elements so that we can also add elements and move $[x, y]$ to map to $[a_1, b_1]$. We may need to do this several times as we do not know $|[a, b]|$. Actually, this argument can be simplified by the following result of Moses [157].

Theorem 4.19 (Moses [157]) *If A is a computable linear ordering with computable block relation, A is isomorphic to a 1-computable linear ordering.*

Sketch Proof Let A be a computable linear ordering with a recursive block relation. Build 1-computable $B \cong A$. In view of Theorem 3.23 it suffices to construct B with computable successivities. Now at stage 0 we map a_0 to b_0. Suppose at some stage s we see x in a_0's block with $x < a_0$, say. Map b_s to x and declare b_s to be b_0's predecessor. If we see at some $t > s$ some z occur with $x < z_a < a_0$ we *change the maps* and set $b_s \to z$ and $b_t \to x$. Since x *is in the same block* as a_0 we'll need to do this at most finitely often and the above can be assembled to give a Δ_2 isomorphism from B to A. □

[*Technical aside re Theorem 4.19* As we mentioned previously it is possible to construct a 1-computable A not isomorphic to any computable B with recursive block relation (Moses). Furthermore although A is *isomorphic* to some 1-computable is with recursive block relation, A itself may not have computable successivities. Of course if A is computably categorical it must be 1-computable.]

Proof of (4.19) continued The result above clearly simplifies case 2a since we can assume A is a 1-computable linear ordering and hence know the size of the interval $[a, b]$. This fact similarly helps with case 2b below.

Case 2b

ra$h$ is bounded. Suppose ra$h = \{n_0, \ldots, n_k\}$, where these all occur infinitely often. Here we can take $n_0 < n_1 < \cdots < n_k$. Since A is not of the correct order type, there must be either an ω or ω^* sequence of pairs of blocks (p_i, q_i) with $|p_i| < |q_i|$. Using this fact we can again use Remmel 'isomorphism changing' to start with a block we assume to be of size p_i in A and then kill φ_e by making its pre-image of size q_i. Using the usual priority methods means that we will settle down on the correct size. □

We remark that Schwartz's result can be obtained by using Theorem 4.16 together with Theorem 4.19.

Another area some authors have examined is that of *discrete* linear orderings. The reader will recall that these are the linear orderings where each element has a successor and a predecessor. nrom section 3 the reader will recall that discrete linear orderings have special properties *since a discrete linear ordering is 1-computable iff it is uniformly n-recursive for all n and hence decidable.*

We shall further examine discrete linear orderings later, but for this section concern ourselves with only one set of results.

Let **D** denote the class of (order types of) decidable discrete linear orderings, **BD** the subclass of **D** with (additionally) the block relation computable and **SBD** the subclass of **BD** with the relation "x and y are in adjacent blocks" computable. We can use Feiner-type coding arguments to show $\mathbf{D} \not\subseteq \mathbf{BD} \not\subseteq \mathbf{SBD}$. However, as we shall see, they have a much more precise relationship.

If $\mathbf{A}$ is a discrete linear ordering then $\mathbf{A} \cong \zeta\rho$ for some order type ρ. We call ρ the *finite condensation* of $\mathbf{A}$ (Rosenstein[195]) and write $C_F(\mathbf{A}) = \rho$. It is easy to see the following:

Observation 4.20 (Moses [160]) *(i) $\tau \in \mathbf{BD}$ iff $C_F(\tau)$ has a computable copy.*

(ii) $\tau \in \mathbf{SBD}$ iff $C_F(\tau)$ has a 1-computable copy.

The main result of Moses [160] is

Theorem 4.21 (Moses [160]) *$\tau \in \mathbf{D}$ iff $C_F(\tau)$ is isomorphic to a Π_1 linear ordering.*

Proof We prove this in some detail, as we will need to modify it for the more difficult arguments to come. The nontrivial direction is to show that if $C_F(\tau)$ is Π_1 then $\tau \in \mathbf{D}$.

Let $\mathbf{A}$ be a Π_1 linear ordering with $x \in A$ iff $(\forall x)(R(x, s))$ and $A \leq \mathbb{Q}$. We use the *amalgamation of blocks* technique. We have $A = \cap_s A_s$ with each A_s computable. This follows since we can regard $x \in A_s$ iff $(\forall t < s)(R(x, t))$. Let $A_s = \{a_{0,s}, a_{1,s}, \ldots\}$ in order of Gödel numbers.

We begin the construction at stage 0 by putting a point z on each side of $a_{i,s}$ for all $i \in \omega$. This defines a block $B(a_{i,s}, 0) = za_{i,s}z$. (To avoid confusion, we will call such new points z-points. An a_i-point can become a z-point but not conversely.) The rough idea is that at stage 1 we add another pair of z-points to each of these blocks to get (roughly) $B(x, 0) = zxz$ with $B(x, 1) = zz\,x\,zz$ and so forth to give $\omega + \omega^*$ around x.

The only trouble at some stage s is if x dies. That is we see $\neg R(x, s)$ hold. Now we want to *stop* building around x and get rid of what we have done. We can't exactly do this, but we can incorporate our work for x into an adjacent block. (Pick the closest and if there is a choice pick e.g. the right one.) So

$$\underset{x\text{-block}}{zzz\,x\,zzz} \quad \underset{x'\text{-block}}{zzz\,x'\,zzz}$$

at stage 3 perhaps becomes the amalgamated block

$$zzzz\,zzz \quad \underset{x'\text{-block}}{zzz\,x'\,zzzz}$$

at stage 4. Note that x has become a 'z' point.

To see that the resulting ordering is decidable, it suffices by Langford's result to see that the successivity function is computable. But this is easy. For a pair x, y present at stage n (x, y) is a successivity at stage $n + 1$, iff (x, y) is a successivity. Evidently the order type that results is correct, giving the proof. $\square$

5 Theorems of Watnick and of Lerman

In this section we shall introduce a quite fundamental proof technique: the infinite injury (Π_2-) priority method. We will do so via a proof of a theorem of Watnick [224] which evolved from some earlier and simpler work of Lerman [130]. We will look at the exact context of Watnick's and Lerman's original results later in Section 6, but here we will be concerned with extending the Moses results of the end of Section 4 (e.g. Theorem 4.21) to computable orderings.

First we shall begin with some (possible obscure) general comments. In the finite injury priority method we must build an object C to satisfy certain conditions which are arranged in a priority list

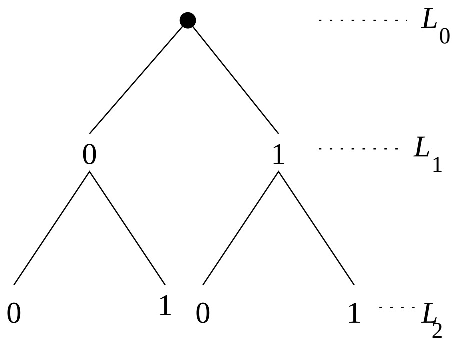
$$L_0,\ L_1,\ L_2,\ \ldots$$

The problem is that we don't know if we ever need to act for the sake of L_i and that an L_i-action can injure all L_j-actions for $j > i$. However if we can arrange matters so that each L_i only acts finitely often the L_j will eventually reach an environment where it can be met since it will be protected by the priority orderings.

In a typical infinite injury argument, the situation is more complex. Now we have a priority list

$$L_0,\ L_1,\ \ldots$$

where some L_i's can act *infinitely often*. Hence at no stage in the construction can we afford to pretend L_i has *finished acting*. Usually we will have several versions of L_j, for $j > i$, guessing L_i's behaviour. We can also do this for a finite injury argument as follows. Here, for simplicity, suppose that if L_i is not injured, it will act at most once. The strategy for the L_i can be thought of as a binary tree



We interpret 1 as 'L_i does not act' and 0 as 'L_i acts'. Thus the string $1\widehat{phantom}\eta1\widehat{phantom}\eta0$ encodes 'L_0 and L_1 don't act' and 'L_2 acts'. In the course of the construction we have a path δ_s with $\lim_s |\delta_s| \to \infty$ and δ_s moving left. For instance, if the construction appears as $1\widehat{phantom}\eta1\widehat{phantom}\eta0\widehat{phantom}\eta1$ at stage 3 and then L_1 acts, the correct next string would be $1\widehat{phantom}\eta0\widehat{phantom}\eta1\widehat{phantom}\eta1\widehat{phantom}\eta1$. Note that we have 1 not 0 in the L_2 position since L_2 has not acted *in a manner consistent with the guess* $1\widehat{phantom}\eta0$ (i.e. after L_1 acts). This corresponds to L_2 being 'injured' by L_1 acting. The finite injury method is often nowadays called the Π_1-method since we only need a Π_1-oracle to figure out exactly how we meet the L_i, that is, to figure out the *true path* (TP) of the construction on the tree. This is defined as the left most path visited infinitely often: if $\sigma \subset TP$ then $\sigma\widehat{phantom}\eta0 \subset TP$ iff $(\exists^\infty s)(\sigma_s \subset TP)$ otherwise $\sigma\widehat{phantom}\eta1 \subset TP$.

In the Π_2 method, the requirements are such that (using the tree model above) the path δ_s can move both left and right. We must ensure that the *versions* of the L_i on the tree path have the correct environment to be met. Plainly this only works with certain requirements. The keys are to get

(i) the 'wrong guesses' not to interfere too much and

(ii) an environment where the 'correct guesses' all cohere.

The following example will hopefully clarify these vague points. It is not difficult to see that if γ has a computable copy then $C_F(\gamma)$ has a Π_2 copy (i.e. is a sub-ordering of $(\mathbb{Q}, <)$) by counting quantifiers (compare with Theorem 4.21) Rosenstein [195] asked if the reverse implication held i.e. if $C_F(\gamma)$ has a Π_2 copy does γ have a computable copy. The following is due to Watnick [225] but was independently and subsequently rediscovered by Downey [45], and Ash, Jockusch and Knight [8]. (See also Roy and Watnick [199])

Theorem 5.1 (Watnick [225]) *If γ discrete then $C_F(\gamma)$ has a Π_2 copy implies γ has a computable copy. That is, ρ is Π_2 presentable iff $\zeta\rho$ is computably presentable.*

Proof The proof uses the 'amalgamating blocks' strategy of Theorem 4.21, but uses Π_2-guessing to control how we amalgamate blocks. The proof is thus an infinite injury argument. Before we give the formal construction, we shall give some motivational remarks to indicate how our construction works.

Let $A \subseteq (\infty, <)$ be a Π_2-copy of $C_F(\gamma)$. Without loss of generality, we may suppose A is infinite. We let $\mathbb{Q} = \{x_1, x_2, \ldots\}$ be a computable enumeration of $\mathbb{Q}$.

A nice representation of A

First we take a nice representation of A as follows. By standard representation of a Π_2 set, we may suppose without loss of generality that $A = \{f(i) \mid \varphi_{f(i)} \text{ is total}\}$ where $\{\varphi_i : i \in \omega\}$ lists the partial computable functions. We replace this representation by a better one (given by a tree which controls strategies). Define a stage s to be a *σ-stage* (for $\sigma \in 2^{<\omega}$) as follows by induction of $\ell h(\sigma)$ (the length of σ):

(i) Every stage s is a $\emptyset$-stage.

(ii) If s is a τ-stage and $\ell h(\tau) = j = f(i)$.

then if $\varphi^s_{f(i)}(y) \downarrow$ where $y = \mu z$ $(z \notin \text{dom}\varphi^t_{f(i)}(y)$: t is a τ-stage and $t < s)$ we say s is a $\widehat{\tau phantom}\eta 0$-stage. Otherwise s is a $\widehat{\tau phantom}\eta 1$-stage.

This gives us our best representation of A. Namely, A is the 'true path' of the above tree: let β be the true path, that is β is the leftmost path visited infinitely often so that $\emptyset \subset \beta$ and if $\sigma \subset \beta$ we have $\widehat{\sigma phantom}\eta 0 \subset \beta$ iff $\exists^\infty s$ (s is a $\widehat{\sigma phantom}\eta 0$-stage) otherwise $\widehat{\sigma phantom}\eta 1 \subset \beta$. Then

$$A = \{j : \ell h(\sigma) = j \text{ and } \widehat{\sigma phantom}\eta 0 \subset \beta\}.$$

One nice property of this representation is that if $j_1, \ldots, j_n$ is any finite subset of A then $j_1, \ldots, j_n$ all 'appear to be in A' *together* infinitely often. The above representation trick was first noticed by Jockusch [95].

Intuitively, we say that x_j *appears to be in A at stage s* if s is a σ-stage and $\ell h(\sigma) = j$. We define $\sigma_s =$ the unique string with $\ell h(\sigma_s) = s$ and s is a σ_s-stage. Then we say that A *appears to be* $\{j : \tau \widehat{phantom} \eta 0 \subseteq \sigma_s$ *and* $\ell h(\tau) = j\}$ *at stage* s and formally define A_s to be this. For simplicity, we shall also ask that $\forall s\ (x_0 \in A_s)$.

A good model for $\quad (B, <) = (C_F^{-1}(A), \leq)$

We must perform three basic tasks:

(i) Around those points $x_i \in A$ we must build $\omega^* x_i \omega$,

(ii) we must 'incorporate' all those 'bad' $x_j \notin A$ somehow into such blocks and

(iii) we must not, ourselves, build anything else.

A good model for B is given as follows: at stage s we have a set of balls with various markings on them, arranged in a line. We have a supply of new balls we must add to this line, either inserted or added to the ends. These new balls will be z-balls, y-balls, or x_i-balls. The intention is that z-balls are attempting to be part of an ω^*-block, y-balls part of an ω-block and x_i-balls part of A. Later we may change our minds and convert y to z or x_i to y or z. *However, if an x_i-ball turns into a y-ball it can't change back.*

The line of balls we refer to as the 'surface'. An x_i ball on the surface will be marked with a *guess* $\sigma \in 2^{<\omega}$ where $\ell h(\sigma) \geq i + 1$. If there is a stage s where this guess 'proves wrong' we turn this x_i ball in to a y or a z ball (with no guess).

The x_i-balls we must place on the surface at stage s are simply those that appear to be in A at stage s. Roughly speaking, we must place the y's and z's around the x_i's that appear in A at s.

The strategies

To satisfy our three aims we must have a reasonably complicated strategy dictating where to place our new balls at each stage. This strategy must overcome several problems whose solutions we outline below:

Incorporations $\quad$ As a first approximation let us suppose that we have three points which appear in A at stage s in the order

$$x_j < x_i < x_k.$$

Now suppose that *really* $x_j, j_k \in A$ and $x_i \notin A$. What we shall know is that x_j and x_k appear in A together infinitely often, but x_i only appears to be in A finitely often. For simplicity let us suppose that x_j and x_k are successors in A. Thus $\forall p\ (x_j \leq p \leq x_k$ and $p \in A \to p = x_j$ or $p = x_k)$ we then see that locally $B = C_F^{-1}(A)$ should be one copy of $\omega^* + \omega$ at x_j and $\omega^* + \omega$ at x_k with x_i not there. To achieve this, we shall incorporate x_i into x_j's block. (The entire construction is 'left justified'). We do this as follows. At stage s, when x_j appears to be in stage s, we must add one z before x_j and one y following x_j. As it stands, we put z immediately before x_j but put y as far right as possible to be consistent with our current picture of A at stage s.

That is to place y for the sake of x_j we go as far right as possible until we see an x_p also appearing in A (or get to the end of A_s) we then repeat the process to x_p.

For example a typical situation might be

$$\begin{array}{rl} zzz\ x_j\ yz\ x_i\ yz\ x_k\ \cdots x_p \cdots & \text{at stage } s-1 \\ zzzz\,\underbrace{xj\ yz\ x_i\ yz\ x_k\ \cdots x_p}_{\text{no change}}\,yz\ x_p \cdots & \text{at stage } s \end{array}$$

Here, at stage s it appears that $x_j \in A_s$ and x_p is the next 'apparent member' of A_s. Now this idea will work since it will ensure that since x_j and x_k appear in A *together* infinitely often we build infinitely many z's before x_j and similarly before x_k. Also, since x_i appears in A only finitely often we build only finitely many y's and z's between x_j and x_k and thereafter almost always incorporate x_i in x_j's ω-block whenever x_j and x_k appear in A together. Thus this builds

$$\omega^* x_j \underbrace{yy\cdots yzz\cdots z}_{\text{finite}} x_i \underbrace{yprocedurey\cdots y\cdots\omega^*}_{\omega} x_k \cdots$$

and so $\omega^* x_j \omega\omega^* x_k$ as required. We call this strategy 'incorporation' as in Theorem 4.21, since we try to ensure that if x_i doesn't appear in A infinitely often, then x_i gets incorporated into somebody's block.

The problem. Thus far we have discussed how we attempted to place the y- and z-balls. The crucial property that will allow this strategy to work is that between any two *successive* points of A (e.g. like (x_j, x_k)) we must ensure that the wrongly placed 'false points' like x_i can be turned into an ω-ordering (i.e. incorporated into our blocking).

Now perhaps infinitely often there will be new x_t occurring where perhaps $x_j < x_t < x_k$ in the $\mathbb{Q}$-order and x_j, x_t, x_k all appear in A when x_t appears in A, say at stage $s\ (x_t) = s_t$. Since, at such a stage s_t it appears that $x_j, x_t, x_k \in A$, it seems reasonable that we should put x_t between x_j and x_k. The crucial idea now is how we can relate such x_t to x_i.

From some point on it will never seem that $z_i \in A$ so it seems unreasonable that we should put anything between x_j and x_i. That is, although in the $\mathbb{Q}$-order perhaps $x_j < x_t < x_i$, if it doesn't seem that $x_i \in A$ at a stage when we must place an x_t-ball on the surface, we shall place x_i beyond x_t in the following order $x_j < x_i < x_t < x_k$ (as x_k appears in A at the same time).

Thus assuming we are working to the right of x_0, our guiding principle is that we always try to put new x_i's as far right as we can.

This brings us to a minor point. When a new x_i appears on the scene, we first determine if $x_i < x_0$ or $x_0 < x_i$. If $x_i < x_0$ we work as above but everything goes left. *Thus, without loss, we shall assume that x_0 is the least member of A and only work right of x_0.* This simplifies presentation.

Summarising so far, our key idea is that we don't build between any x_p and x_q if x_q is x_p's current successor and x_p doesn't appear to be in A at stage x. This idea helps us to overcome the problems induced by our strategy above.

The next situation we must consider is the situation when x_t, x_j, x_i, x_k above are *all in* A but infinitely often it appears that $x_j \in A$, $x_i \notin A$ and $x_k \in A$, and infinitely often it appears that $x_j \in A$, $x_i \notin A$ and $x_t \notin A$.

Following our strategy above, perhaps at stage s_0 when x_t appeared for the first time we set down the balls in the order

$$x_j \; x_i \; x_t \; x_k \qquad \text{(at stage } s_0\text{)}$$

when their 'real order' is $x_j \; x_t \; x_i \; x_k$. We do so since it appears that $x_i \notin A$ at stage s. But later we see that $x_i \in A$ and we see also x_j, x_t and $x_k \in A$ at some stage $s_1 > s_0$. We now realise that our initial guess as to x_t's position appears wrong and now put

$$x_j \; x_t \; x_i \hat{x}_t \; x_k.$$

Here $\hat{x}_t$ denotes a group of balls around x_t's 'old position'. Now we can't remove these balls but also don't wish to build an $\omega^* + \omega$ block around $\hat{x}_t$, so our solution is to relabel the $\hat{x}_t$-balls as y- or z-balls.

This then gives rise to another problem. Later we again see $x_t \in A$ but now $x_i \notin A$. We can't use the x_t position between x_j and x_i since we really only put it there in the first place since x_j and x_i appeared in A. Perhaps in another triple $x_i \notin A$ and $x_t \in A$ but infinitely many such x_t get inserted between x_j and x_i. We really should put new x_t as far right as we possibly can consistent with our current picture of A_s. If, later, it appears that x_j and x_i are in A, we again build around the x_t we put between x_j and x_i last time and cancel its current position. This leads to the fundamental idea of the section.

Labelling. Whenever we put an x_t on the surface into a new position for the first time we give it the *label* $\sigma \subseteq \sigma_s$ with $\ell h(\sigma) = t + 1$. Thus we are saying: here is where x_t's correct position is, should σ be correct. Now, if we must move x_t, we do so because some x_i encoded in this guess, which appeared to be *out* of A, now appears to be *in* A, and $i < t$. (Of course, in σ_t this will appear as $\tau\widehat{phantom}\eta 1$ with $\ell h(\tau) = i$.) The condition $i < t$ is simply to sort out which ball to move. When we move x_t, we give its new position a 'better' label. The correct position for x_t is a stable position *corresponding to the leftmost label visited infinitely often.* The reader should note that, at any particular time, x_t might have several positions labelled on the surface. Only one (at most) is correct. Here we need another notion. Let $\leq_L$ denote the lexicographic ordering on the tree. Thus $\sigma \leq_L \tau$ iff $\sigma \subseteq \tau$ or $(\exists\gamma)$ $(\gamma\widehat{phantom}\eta 0 \subseteq a$ and $\gamma\widehat{phantom}\eta 1 \subseteq \tau)$. The reader should read $\sigma \leq_L \tau$ as $\sigma = \tau$ or σ is *stronger* than τ. Note that only those $\sigma \leq_L \tau$ will always have there apparent positions uncancelled. Those $\tau \leq_L \sigma$ only get visited finitely often and so only move σ finitely often. Hence we shall argue that σ reaches a stable position and so does x_t. We now give the formal details of the construction, although we hope that readers can see them for themselves.

Construction, stage $s + 1$

Construction of $C_F^{-1}(A) = B$, say stage $s + 1$

Step 1 (Cancellation) Compute σ_s. Cancel all positions marked τ for $\tau \not\leq_L \sigma_s$. Regard these now as y-balls and similarly any z-balls associated with them become y-balls.

Step 2 (Placing x_j-balls) In order of j, for each τ with $\ell h(\tau) = j$ and $\tau\widehat{phantom}\eta 0 \subseteq \sigma_s$, proceed as follows:

If there is currently an (uncancelled) position marked $\tau\widehat{phantom}\eta 0$ on the surface, do nothing. If there fails to be such a position, establish one by placing an x_j-ball marked $\tau\widehat{phantom}\eta 0$

as far right as possible. This will be on the right of B_{s-1} unless there is an x_i-ball marked $\gamma\widehat{phantom}\eta 0 \subseteq \tau\widehat{phantom}\eta 0$ and $x_j < x_i$. In this case, for the $<_{\mathbb{Q}}$-least such x_i, we place x_j immediately left of x_i and put s z-balls immediately preceding x_i. $(x_j zzz \cdots zx_i)$.

Step 3 (Placing y, z-balls). Now place y- and z-balls as indicated in the discussion. That is place another z-ball before x_0. Now go right and find the first x_j-ball, if any, marked $\gamma\widehat{phantom}\eta 0 \subseteq \sigma_s$ for some γ. If there are no y- or z-balls between x_j and its predecessor, add a y- followed by a z-ball. If there are, already, a block of y's and a block of z's preceding x_j, add one further y to the y-block and one further z to the z-block. Now continue until we get to the right end. Here, add one y-ball.

End of construction

Verification: For the sake of the following lemmata we shall adopt the following definitions:

Definition 5.2 We say a *ball n appears to be in an x-ball g's ω^*-block at stage s* if n is a z-ball and, if m is any ball with m between n and g then m is a z-ball.

Definition 5.3 We say a *ball n appears to be in an x-ball g's ω-block at stages* if g has guess σ for some $\sigma \subseteq \sigma_s$ (and g is not cancelled at s), $g \leq n$, and one of the following conditions is satisfied:

- (i) (a) there exists an x-ball g in domB_s with $g \leq n < \hat{g}$ such that $\hat{g}$ has guess $\subseteq \sigma_s$ such that
  - (b) there does not exist an x-ball in domB_s with guess $\subseteq \sigma_s$ and $g < r \leq n$ and
  - (c) there exists a y-ball p in domB_s such that $n \leq p < \hat{g}$, or
- (ii) there does not exist an x-ball $\hat{g}$ in domB_s with $g < \hat{g}$.

Remark 5.4 The intuition here is that either n occurs beyond the largest apparent member of A at s (in 2) or n occurs before the z-block of two apparently consecutive (at s) elements of A at stages.

We also say that position g (an x-ball) *receives attention at stage s* if the number of elements in g's apparent ω^*-block at s increases.

Let β denote the leftmost path. That is, as in the discussion, β is the leftmost path visited infinitely often.

Lemma 5.5 (Stable position lemma) *Every x_i reaches a stable position. That is, either $x_i \notin A$ in which case x_i receives attention finitely often (in total) or $x_i \in A$ in which case there is a* **unique** *x_i-ball that receives attention infinitely often (and is, of course, never cancelled). This ball is marked $\sigma\widehat{phantom}\eta 0 \subseteq \beta$ with $\ell h(\sigma) = i$.*

Proof By induction. Suppose for all $j < i$ the lemma holds. Let s_0 be a stage such that $\forall s_0\ (\sigma\widehat{phantom}\eta 0 \leq_l \sigma_s)$.

Let $s_1 > s_0$ be the least $\widehat{\sigma phantom m \eta 0}$-stage exceeding s_0. Now at stage s_1, choosing s_0 minimal, we may suppose that we place an x_i-ball h, marked $\widehat{\sigma phantom m \eta 0}$. We claim that this is x_i's stable position.

First, this position cannot be cancelled, we only cancel positions when their guess appears wrong (in step 1 of the construction). By 1 above this cannot occur. Hence this position is never cancelled.

Second this position receives attention infinitely often since $\widehat{\sigma phantom m \eta 0} \subset \beta$ and so $\exists^\infty s(\widehat{\sigma phantom m \eta 0} \subseteq \sigma_s)$ by definition of β.

Finally h_i is unique. To see this let g be another x_i ball that receives attention infinitely often. This ball g must have guess γ, for some γ with $\ell h(\gamma) = i+1$. There are three possibilities: either $\gamma \leq_L \beta$ and $\gamma \not\subseteq \beta$, $\gamma \subset \beta$ or $\beta \leq_L \gamma$. In the first case, there are only finitely many γ-stages. We only add to this x_i-ball's z-block at γ-stages (by construction) and so such an x_i-ball can receive attention at most finitely often.

If $\gamma \subset \beta$ then $\gamma = \widehat{\sigma phantom m \eta 0}$. There are two possibilities. Either g was appointed at a stage t *before* x_i (i.e. before s_1) or g was appointed *after* h, so at a stage $s_2 > s_1$. In the first case our assumptions concerning the minimality of s_1 mean that there is a η-stage $\hat{t}$ with $t \leq \hat{t} \leq s_0$, with $\eta \leq_L \gamma$ and $\eta \neq \gamma$. Such a stage cancels g. If g is appointed after stage s_1 then g was appointed at a $\widehat{\sigma phantom m \eta 0}$-stage. We only appoint new positions if there is not already a $\widehat{\sigma phantom m \eta 0}$-position available. There is, of course, namely the one occupied by h. Thus we wouldn't have appointed g after all.

Finally, if $\beta < \gamma$ then $\widehat{\sigma phantom m \eta 0} \not\subseteq \gamma$. By step 1, at each $\widehat{\sigma phantom m \eta 0}$-stage we cancel any x-balls marked γ. Hence g gets cancelled and so wouldn't have received attention after all. $\square$

Lemma 5.6 (True z-ball lemma) *Let $x_i \in A$. Let $\hat{x}_i$ denote x_i's stable position (given by Lemma 5.5). Suppose n is a ω^*-ball of $\hat{x}_i$ at stage s_1. Then n is a z-ball of $\hat{x}_i$ at every $\widehat{\sigma phantom m \eta 0}$-stage where $\widehat{\sigma phantom m \eta 0} \subseteq \beta$ and $\ell h(\sigma) = i$. Also if $z(n, s)$ is the number of balls between n and $\hat{x}_i$ at stage s then for all $s \geq s_1$, $z(n, s) = z(n, s_1)$ $(= z(n)$, say).*

Proof Let n, $\hat{x}_i$ be as above at stage s_1. By definition of ω^*-ball, n is a z-ball and there are no balls between n and $\hat{x}_i$ save for z-balls. The construction (in step 2) specifically ensures that when we place new x-balls $x_j < \hat{x}_i$ we do not disrupt any currently placed z-balls. In particular, no new γ_j balls can be placed between n and $\hat{x}_i$. This, of course, means that $z(n, s) = z(n, s_1) = z(n)$ since new z-balls are always placed on the *left* end of $\hat{x}_i$'s apparent ω^*-block. $\square$

Lemma 5.7 (True ω^*-block lemma) *Let $x_i \in A$ and $\hat{x}_i$ as in Lemma 5.6. Then in B there appears an ω^* $\hat{x}_i$-block.*

Proof By Lemma 5.6 and the fact that $\hat{x}_i$ receives attention infinitely often, and hence we add infinitely many z-balls before $\hat{x}_i$. $\square$

Definition 5.8 Suppose n is a z-ball of some $\hat{x}_i$. Then we say n is a *stable ω^*-ball,* and we say n *adheres to* $\hat{x}_i$ (as an ω^*-ball).

Lemma 5.9 (ω-adherence lemma) *Let m be any ball and suppose m is not a stable x_i-ball or a stable ω^*-ball. Then there is a stable x_i-ball $\hat{x}_i$ and a stage $s(m)$ such that*

(i) m appears to be an ω-ball of x_i at each $\widehat{\sigma phantom m\eta}0$-stage $> s(m)$ where $\widehat{\sigma phantom m\eta}0 \subseteq \beta$ with $\ell h(\sigma) = i$.

(ii) If $d(m, \hat{x}_i, s)$ denotes the number of balls between m and $\hat{x}_i$ at stage s then $\forall s > s(m)$ $(d(m, \hat{x}_i, s) = d(m, \hat{x}_i, s(m)) = d(m, \hat{x}_i))$.

(In the case above we say m adheres to x_i as an ω-ball.)

Proof This is the main lemma that our machinery – discussed before the construction – is meant to achieve. Now whether m is an unstable z-ball, an unstable x_j-ball for some j or a y-ball. All of the cases are similar and can, roughly speaking, be treated simultaneously. Let s be the stage when m was placed on the surface. If m was an x_j-ball let $c(m) = m$. If m was a y-ball, find the $\leq$-greatest x-ball x_j alive at stage s with $x_j \leq m$ and set $x(m) = x_j$. (Note: x_j *doesn't* need to be apparently in A at s. Also x_j may later die or x_j may be $\hat{x}_j$.) Finally if m was a z-ball, find the $\leq$-least x-ball x_j with $m \geq x_j$ and set $c(m) = x_j$. Note that m appears to be in x_j's z-block at stage s. Since there are no dormant x-balls between m and $c(m)$, as in Lemma 5.6, we cannot place more balls between $c(m)$ and m after stage s by the way we place balls. It therefore suffices to argue the lemma for $x(m) = x_j$ instead of m.

Without loss, $x_j = c(m)$ is unstable, and either x_j is eventually cancelled or x_j is never cancelled but only appears in A finitely often.

Now let $\hat{x}_j$ be the $\leq$-greatest stable position $\leq x_j$ in B *at stage s*.

Claim 5.10 *After stage s we cannot add a stable position $\hat{x}_t$ with $\hat{x}_i < \hat{x}_t < x_j$.*

Suppose not and $\hat{x}_t$ is such. Let $\widehat{\sigma phantom m\eta}0$ be the guess of x_t. The only time we place such a ball $\hat{x}_t$ between balls already on the surface is because the guess forces us to. That means there must be some ball x_q with guess $\widehat{\gamma phantom m\eta}0$, say and $\hat{x}_i < \hat{x}_t < x_q < x_j$ forcing $\hat{x}_t < x_j$. Since this is so, by priorities of movement it must be that $q < t$.

For suppose otherwise and $t < q$. Now x_q must be already present on the surface at the stage s when x_i enters. Since this is a *new* position for x_t there must be no position marked $\widehat{\sigma phantom m\eta}0$ on the surface at stage s. Hence it cannot be that the guess $\widehat{\gamma phantom m\eta}0$ of x_q extends $\widehat{\sigma phantom m\eta}0$, because if this was so we would already have put down a x_t-ball g marked $\widehat{\sigma phantom m\eta}0$ by the time we put $\widehat{\gamma phantom m\eta}0$ down for z_q. But then this ball g cannot have been cancelled in the intervening stages since if g were cancelled, so too x_q would have been cancelled. (g would be cancelled since $\widehat{\sigma phantom m\eta}0 \not\leq_L \sigma_s$ but then as $\widehat{\sigma phantom m\eta}0 \subseteq \widehat{\gamma phantom m\eta}0$, $\widehat{\gamma phantom m\eta}0 \not\leq \sigma_s$). Hence we see $q < t$.

Now as $q < t$ and $\hat{x}_t$ is a stable position we must have that x_q is also a stable position. (Remember, in this case $\widehat{\sigma phantom m\eta}0 \subseteq \widehat{\gamma phantom m\eta}0$ and $\widehat{\sigma phantom m\eta}0 \subseteq \beta$. If x_q's position is cancelled, so too is anything, in particular $\hat{x}_t$, marked $\widehat{\sigma phantom m\eta}0$.)

In either case we see that no such x_q (and hence $\hat{x}_t$) can exist, giving (5.10).

Thus we have Claim 5.10 that there are no stable positions between $c(m) = x_j$ and $\hat{x}_i$. We now show that x_j adheres to $\hat{x}_i$.

Let $x_{i_1}, \ldots, x_{i_n}$ list those x_0-balls alive at stage s with $\hat{x}_i < x_{i_1} < \cdots < x_{i_n} = x_j$. Let $s(m)$ be the least stage such that for all j with $1 \le j \le n$ we have

(i) either x_{i_j} is cancelled at stage $s(m)$, or

(ii) $\forall s > s(m) \quad (\gamma_j \not\subseteq \sigma_s$ where γ_j is the guess of $x_{i_j})$.

Such a stage must exist by Claim 5.10 and the definition of stability. As the notation suggests, we claim that this $\hat{x}_i$ $(s(m))$ is correct, namely that

Claim 5.11 *(i) x_j appears to be an ω-ball of $\hat{x}_i$ at each $\widehat{\sigma phantom\eta}0$-stage $> s(m)$.*

(ii) $d(x_j, \hat{x}_i, s(m)) = d(x_j, \hat{x}_i, s)$ for all $s > s(m)$.

By construction, $2 \Rightarrow 1$ because of the way we place y-balls and the fact that $\hat{x}_i$ is stable. We argue that 2 holds in a similar way to our argument that there are no stable x-balls between x_j and $\hat{x}_i$. Suppose $d(x_j, \hat{x}_i, s(m)) < d(x_j, \hat{x}_i, s)$ for some least $s > s(m)$.

There are two ways we might insert new balls between x_j and $\hat{x}_i$. Either the new ball n is a y- or a z-ball placed between $\hat{x}_i$ and x_j because we see that x_q appears in A at s (i.e. it's guess appears correct) for $\hat{x}_i < x_q < x_j$ (notice that $x_q < x_j$ and $q \neq i_k$ for any $1 \le k \le n$ by choice of $s(m)$) or the ball is a new x-ball x_d, say, placed there because again we see x_q whose guess appears correct with $\hat{x}_i < z_q < z_j$. Again $x_q \neq x_j$ or x_{i_k} for any $1 \le k \le n$. We claim that in either case no such z_q can exist.

Now, using the reasoning of Claim 5.10, x_q which did not exist at stage s must have appeared at a stage where some of the x_{i_k} for $i \le k \le n$ looked correct, since otherwise we'd placed it beyond x_j. Also using the same reasoning as Claim 5.10 it must be that q exceeds those x_{i_k} that forced it in (otherwise x_{i_k} would be cancelled first) and so since x_q's guess *extends* such x_{i_k}'s guess, x_q's guess appears correct at best only when x_{i_k}'s does too. But now, choice of $s(m)$ means that x_{i_k}'s guess never again looks correct. Thus x_q's guess also never again looks correct. Hence stage s can't exist after all. This clinches 2 and hence Claim 5.11 and the lemma follows. □

Lemma 5.12 (Truth of outcome lemma) *Let n be any ball. Then either n is a stable x-ball or n adheres to a stable x-ball.*

Proof By Lemmas 5.7 and 5.9. □

Lemma 5.13 $C_F(A) \cong B$.

Proof The desired isomorphism is induced by the injection from $A \to B$ given by $\hat{x}_j \to x_j$. The lemmata and addition of y points achieve the rest. □

As we see in Sections 6 and 8, Watnick's theorem has a number of important extensions and generalisations. Along related lines, Lerman [130] also used an infinite injury argument controlled by a tree to show:

Theorem 5.14 (Lerman [130], Theorem 2.2) *If A is an infinite Σ_3 set, then there is a computable linear ordering B of order type*

$$\zeta + n_0 + \zeta + n_1 + \cdots$$

with n_0, n_1, ... listing A in order of magnitude.

(Here, and henceforth, we assume $0 \notin A$.)

We shall call the representation of A in Theorem 5.14 a *strong ζ-representation*. If this representation is not in order and perhaps has repetitions, we shall call it merely a *ζ-representation.* Note that by Theorem 5.14 and counting quantifiers, A has a strong ζ-representation iff A has a ζ-representation iff A is Σ_3.

We can similarly define (strong) η-representation. These have been investigated by Lerman [130], Feiner [64], Fellner [62] and Rosenstein [195].

Theorem 5.15 *(i) (Feiner [64]) If A has an η-representation the A is Σ_3 (see Theorem 3.7).*

(ii) (Rosenstein [195]) If A has a strong η-representation then A is Δ_3.

(iii) (Rosenstein [195])Every Σ_2 set has a strong η-representation.

(iv) (Fellner [62]) Every Π_2 set has a strong η-representation.

Proof (1) and (2) come from looking at the quantifier form. For instance for (2), by (1) we know if A has a strong η-representation L then A is Σ_3. So it suffices to show that it is also Π_3. We need only show $\bar{A}$ is Σ_3 for this. Now $n \in \bar{A}$ iff there are $x_1, \ldots, x_{n+1}$ with (x_i, x_{i+1}) a successivity if $i \neq j$ such that for all distinct $y_1, \ldots, y_n$ with all the $y_i \leq x_1$ for all i there is an i, j and a z with $y_i < z < y_j$. Clearly this is Σ_3 and defines $\bar{A}$ as L is a *strong* representation.

(3) Let A be Σ_2. So for some computable R we have $x \in A$ iff $(\exists y)(\forall z)(R(x, y, z))$. By a standard trick we can choose R so that there is $X \in A$ iff there is a *unique* y such that for all z, $R(x, y, z)$. We then construct intervals $I_0 < I_1 < \cdots$ so that the desired ordering is $\sum_x I_x$. We subdivide I_x into intervals $\{J_{<x,i>} : i \in \omega\}$ so that $I_x = \sum_y J_{<x,y>}$. $J_{x,y>}$ will either have order type x or η. We start with $J_{<x,y>}$ of type x until we see some z with $\neg R(x, y, z)$. We then make $J_{<x,y>}$ have type η. This evidently works.

The proof of (4) can obviously be assembled along lines similar to Watnick's or Lerman's by taking a Π_2 A represented by a tree and then using this to place points. We sketch this below.

Let A be Π_2 so that $n \in A$ iff $(\forall x)(\exists y)(R(x, y, n))$ with R the relation representing the true path on a tree. Without loss of generality $|A| = \infty$. We sketch the argument for two numbers. Suppose 2 and 4 are the least two numbers in A. The construction begins by trying to build an ω sequence

$$x_0^s x_1^s x_2^s \cdots$$

When we see n appear to be the least number in A we must chop the above down to n. That is, we will ensure that a dense piece follows a block of n things. As $2 = \mu z$ $(z \in A)$ we need to construct this so that the final type is

$$2 + \eta + \cdots$$

Now 1 may appear in A finitely often. Each time it does so, we can cancel any work we have done based on the belief that $2 = \mu z$ $(z \in A)$. In particular we may cancel the block x_0^s, x_1^s and put points between x_0^s and x_1^s building the beginning of an η-sequence. However at some stage t we will have that, for all $x > t$, $1 \notin A_s$. Then at t we will ensure that $(\forall t_1 > t)(x_0^t = x_0^{t_1}$ and $x_1^t = x_1^{t_1})$.

Now (after t_1) suppose we next see a stage t_2 where $2 \in A_{t_2}$ and $3 \in A_{t_2}$.

Then we will have to attempt

$$x_0 x_1 \eta \cdot z_0 z_1 z_2$$

To do this we block off $z_0,\ z_1,\ z_2$ and agree that

(i) this will be *part of* a block based on the guess $2 \in A$ and $(\forall t_3 \geq t_2)$ $(1, 0 \notin A_{t_3})$, and

(ii) this will be a complete block if *additionally* $3 \in A$.

To make 1 and 2 compatible, we need only ensure that if $3 \notin A$ then $z_0\, z_1$, z_2 is incorporated into some larger block.

Later, as stage u, we may see $2 \in A_u,\ 3 \notin A_u,\ 4 \in A_u$. Then we would densify between $x_0\, x_1$ and the $z_0\, z_1\, z_2$ blocks regarding the point currently following z_2 as part of a 4-block. A typical situation is

$$x_0\, x_1\, p\, p\, p\, p\, p\, z_0\, z_1\, z_2\, q\, q\, q\, q$$

becoming

$$x_0\, x_1\, r\, p\, r\, p\, r\, p\, r\, p\, r\, p\, r\, p\, r\, z_0\, z_1\, z_2\, q\, r\, q\, r\, q\, r\, q\, r$$

where r represents the new points. Then if later we see $3 \in A_u$, we'd add a new point between z_2 and q.

In this way we will build the correct order type corresponding to the true path of T. $\square$

A classification of the η-representable A seems quite hard. The depth of the problem was revealed by the following result of Lerman.

Theorem 5.16 (Lerman [130]) *There is a Δ_3 A such that A is not η-representable.*

Proof The technique here is rather different from previous arguments and is an 'oracle' construction. Let D be Δ_3 complete and L_0, L_1, ... list the computable linear orderings. We ensure that $A \leq_T D$ so that A is Δ_3. The requirements we meet are

$$R_e\ :\ L_e \text{ does not } \eta\text{-represent } A.$$

Let $[a,\, b]_e = \{x \in L_e \,|\, a \leq x \leq b\}$. At stage s, of the construction we perform the following steps:

Step 1 For each $i < s$ which is *active* via an interval $[a_i^s,\ b_i^s]$ at the end of $s-1$, ask D if $[a_i^s,\ b_i^s]$ is a complete block of cardinality s. If the answer is no for all $i < s$, put s into A. If the answer is yes for some i let $i = i(s)$ and put s in $\bar{A}$. Declare i as inactive.

Step 2 If $i(s)$ is defined in step 1, cancel the assignment of $[a_i^s, b_i^s]$ to L_i for all $i \geq i(s)$.

Step 3 Fix the least i with L_i active, yet no interval assigned to L_i. Ask D if L_i is total. If L_i is not total declare L_i as inactive (and go to step 4). Ask D if L_i has points a, b with $s+1$ points between them. If not, declare L_i as inactive and go to step 4. If the answer is yes, fix a finite interval $[a_i^s, b_i^s]$ that is not a complete block and has cardinality s and assign it to L_i.

Step 4 For each $i \leq s$, if there is an interval currently assigned to L_i that is not maximal, use D to construct $[a_i^{s+1}, b_i^{s+1}]_i \underset{\neq}{\supset} [a_i^s, b_i^s]$ of L_i of cardinality $s+1$. This is now assigned to L_i in place of $[a_i^s, b_i^s]$.

By construction $A \leq_T D$. Note that if $i(s)$ is defined then $i(s+1) < i(s)$ and hence $|A| = \infty$. To see that L_i does not η-represent A, for an induction let s_0 be a stage where the L_j for $j < i$ are not activated after stage s_0. Then if $[a_i^s, b_i^s]$ is assigned to L_i after s_0 (choosing s_0 least) we see that as this assignment is not cancelled, if L_i is not inactivated then $\cup_{s>s_0}[a_i^s, b_i^s]$ is an infinite block in L_i, so L_i cannot represent A. On the other hand, if i is inactivated at $t > s_0$ then L_i has a complete block of size t, yet $t \notin A$. The result follows. □

Lerman did note that although not all Σ_3 sets can be realized by η-representations, all such *degrees* can be.

Theorem 5.17 (Lerman [130]) *Let A be Σ_3. Let $B = A \oplus \omega = \{2x : x \in A\} \cup \{2x+1 : x \in \omega\}$. Then B is η-representable. Since $B \equiv_T A$ it follows that all Σ_3 degrees can be η-represented.*

Proof The argument is a simple coding one. If A is Σ_3, we know for some computable R that $x \in A$ iff $(\exists y)(\forall z)(\exists t)(R(x, y, z, t))$. For each $e \in \omega$ we construct an interval I_e and then the required ordering will be $L = \sum_{e \in \omega} I_e$.

Let $e = \langle x, q\rangle$. We build $I_{\langle x, q\rangle}$ as follows. If x is odd we simply make $I_{\langle x, q\rangle}$ of type $\eta + x + \eta$. If x is even we begin by having the interval of type $\eta + x + 1 + \eta$. If we see $R(x, q, 0, s)$ hold for some s, we say that 0 is verified for (x, q) and absorb the 1 into an η piece so that for one stage, $I_{\langle x, q\rangle}$ has type $\eta + x + \eta$. At the next stage, we put a new 1 back in and have $\eta + x + 1 + \eta$ again until 1 is verified i.e. $R(x, q, 1, t)$ also holds for some t. It is easy to see that L has the desired properties. □

Lerman notes that Theorem 5.17 can easily be modified to work for any Σ_3 set B that is not immune. (That is, B has an infinite computable subset.)

Question 5.18 *Is there a similar classification of strongly η-representable Δ_3 degrees? In particular, is each Δ_3 degree strongly η-representable?*

We close this section with a brief look at another special class of order types. Let $f : \mathbb{Q} \to \mathbb{N} - \{0\}$. We shall say an order type $\mathbf{A}$ is *η-like* if it has the form $\Sigma\{f(q) : q \in \mathbb{Q}\}$. We shall say $\mathbf{A}$ is *strongly η-like* if the range of f is finite. Because of their simplicity and because they turn up in many contexts (as we will see and have seen) η-like orderings have attracted quite a bit of attention. We restrict ourselves in this section to studying the complexity of f if $\mathbf{A}$ is a computable (strongly) η-like ordering, and conversely. The first result here is

Theorem 5.19 (Rosenstein [195]) *If* $\mathbf{A}$ *is computable and* η*-like then there is a* Δ_3 *function* g *such that* $\mathbf{A} \cong_{\text{comp}} \Sigma\{g(q) : q \in \mathbb{Q}\}$.

Proof We need to build $g \leq_T \mathbf{0}''$. We do so as follows. Let $q_0, q_1, \ldots$ list $\mathbb{Q}$, and $a_0, a_1, \ldots$ list A. At stage s we define $g(q_s)$. Define $f(q_0)$ as follows: Ask the $\mathbf{0}''$-oracle if there is an $a_i \neq a_0$ that is either the successor or predecessor of a_0. Note that $\mathbf{0}''$ can answer such questions. In this way, we know that all blocks are finite, and we can compute $B(a_0)$, the block containing a_0. Now define $g(q_0) = |B(a_0)|$. Now continue in the obvious way. If $q_1 < q_0$ work below $B(a_0)$ and if $q_0 < q_1$ work above $B(a_0)$. If, for example, $q_1 < q_0$, find the least (by Gödel number) $a_i \notin B(a_0)$ and similarly define $g(q_1) = |B(a_i)|$. Now the density of $\mathbb{Q}$ ensures that this can be extended to an isomorphism and g is Δ_3 by construction. □

Using an approximation argument similar to (eg) Theorem 5.15(3) and essentially the tree representation of a Π_2 set, Fellner showed

Theorem 5.20 (Fellner [62]) *If* f *is* Π_2 *then* $\Sigma\{f(q) : q \in \mathbb{Q}\}$ *is computably presentable.*

Proof Let $x \in A$ iff $(\forall z)(\exists y)(R(x, z, y))$, where $A = \{\langle q, n\rangle : f(q_1) = n\}$ when $R(\langle q_1, n\rangle, z, y)$ holds. (We can suppose that for all $m > n, \neg R(\langle q, m\rangle, z, y)$ and say that $\langle q, n\rangle$ looks correct at stage y.) Remember that if $x, y \in A$ then x and y appear *together* on A infinitely often. Now we define a linear ordering B and a partial function f_s from $\mathbb{Q}$ to the blocks of A so that $f = \lim_s f_s$. Define $f_0(q_0) = n_0$ with n_0 such that $R(\langle q_0, n_0\rangle, 0, y)$ holds for some least y and put down a block of n_0 ones, that are labelled $\langle q_0, n_0\rangle$. This label remains unless for some $s > y$ and some $n_i < n_0$ we see $R(\langle q_0, n_i\rangle, 0, s)$ hold in which case we release the last $n_0 - n_i$ ones (to be put into other blocks). If at some stage $t > y$ we instead see $R(\langle q_0, n_j\rangle, 0, t)$ hold for some $n_j > n_0$ then we begin a block of n_j ones *containing* the n_0 ones (and have $f_t(q_0) = n_j$). It is easy to see that after a finite number of failures, if $\langle q_0, n\rangle$ is the *true* member of A, the n ones put down will be a block that will n ever be split. Of course, for infinitely many s we many build blocks with $n_{i_s} > n$ many ones, but whenever n looks correct, they will be split off from the n block.

Strategies are combined in the obvious ways as in (eg) Watnick's theorem so if we have for $\langle q_1, k\rangle$ guesses as to the correct $\langle q_0, n\rangle$. When these prove wrong, elements can be released to the construction. The only care that is really needed is to ensure that any one that is put down eventually adheres to some block, and this is achieved by using the least element that is around as building materials for blocks. □

Similar arguments show

Theorem 5.21 (Downey) *If* f *is* Σ_2 *then* $\{\Sigma f(q) : q \in \mathbb{Q}\}$ *is computably presentable.*

Proof sketch Now $f(q) = n$ iff $(\exists x)(\forall y)(R(\langle q, n\rangle, x, y)$. We map q_0 initially to the first n_0 such that we see $R(\langle q_0, n_0\rangle, x, 0)$ hold for some x. We keep q_0 mapped to n_0 until we see (if ever) $R(\langle q_0, n_0\rangle, x, s)$ fail to hold at some stage s. We then switch to the least n_i such that for some least x, for all $j \leq s$, $R(\langle q_0, n_i\rangle, x, j)$ holds. If $n_i < n_0$ we release those ones in $n_0 - n_i$. If $n_i > n_0$ we build the block around the n_0 block. As there really is some n with $f(q_0) = n$ we eventually settle on $f(q_0) = n$ for some n. The inductive strategies combine in a straightforward way. □

In view of Theorem 5.19, a natural question is whether the above results can be improved to Δ_3. Lerman and Rosenstein [132] shoed that the answer is no.

Theorem 5.22 (Lerman and Rosenstein [132]) *There is a Δ_3 function f such that $\Sigma\{f(q) : q \in \mathbb{Q}\}$ is not computably presentable.*

Proof The method is an oracle construction like Theorem 5.16. So let $\{L_e : e \in \omega\}$ list all computable linear orderings. We build f computably in $\mathbf{0}''$.

To show that L_e does not represent f, we proceed as follows. First $\mathbf{0}''$ can decide if L_e is total. If this is not true then we need do nothing. So we henceforth assume L_e is total. Now we will ensure that for each n there is at most one $q \in \mathbb{Q}$ with $f(q) = n$. As we go along in the construction we will build blocks. Suppose we have an n block. We ask $\mathbf{0}''$ if L_e has two n-blocks for $m \geq n$. If not we define $f(q) = n$, and $f(r) = n$, some q, r and need do nothing more. If L_e has two m-blocks for $m \geq n$, we will find two such sets of n successivities $B_{e,s}$ and $\hat{B}_{e,s}$. We then use these as follows. At each future stage we ask if $B_{e,t}$ and $\hat{B}_{e,t}$ are complete blocks. When the answer is no, for each that is not complete find (eg) $B_{e,t+1} \underset{\neq}{\supset} B_{e,t}$.

In this way we can eventually get an infinite block in L_e (and so we really do nothing for L_e) or eventually we can find complete blocks B_e and $\hat{B}_e$ with $|B_e| \neq |\hat{B}_e|$ and we have ensured that $|B_e|$, $|\hat{B}_e|$ are not yet in ra f. Then if $B_e < \hat{B}_e$ in L_e we define $f(q) = \hat{B}_e$ and $f(r) = B_e$ for some $q < r$ in $\mathbb{Q}$. This makes the order type of L_e wrong. $\square$

We remark that Theorem 5.22 can be improved to construct f with bounded range so that $\Sigma\{f(q) : q \in \mathbb{Q}\}$ would be strongly η-like.

Theorem 5.23 (Downey) *There exists a Δ_3 f with range $\{1, 2, 3, 4\}$ such that $\Sigma\{f(q) : q \in \mathbb{Q}\}$ is not computably presentable.*

Proof Sketch This time the strategy is different. First we define $f(q) = 4$ for all $q \in \omega$ with $q > 1$. We then define $f(q) = 2$ for all $q \leq 0$ with $q \in \mathbb{Q}$. The intervals between these blocks, we will either make of type $\eta + 3 + \eta$ on η, depending on how we wish to diagonalise against the L_e's. We use the interval between $f(e)$ and $f(e+1)$ to kill L_e. For L_0 we ask $\mathbf{0}''$: does L_0 have a block of 4 points? If yes, find one and ask $\mathbf{0}''$ if it is complete. If yes, call this $B_{0,0}$. If no, forget L_0 and make the interval $(f(0),\ f(1))$ of type η. Now ask $\mathbf{0}''$: does L_0 contain a 4-block to the left of $B_{0,0}$.

(i) If yes, define $f(r) = 1$ for the least (by Gödel number) $r \in (0, 1)$. Find such a block $B_{0,1}$ to the left of $B_{0,0}$ and now repeat the process above, with $B_{0,1}$ in place of $B_{0,0}$.

(ii) If no, ask $\mathbf{0}''$ if L_0 contains a 3-block to the left of $B_{0,0}$ or, more generally, $B_{0,n}$. If yes, it must be a complete 3-block and we can win by making $f(q) = 1$ for all $q \in (0, 1)$. If no, then we can pick some $q \in (0, 1)$, with $f(q)$ as yet undefined, and define $f(q) = 3$ with $f(r) = 1$ otherwise, for $r \in (0, 1)$. This kills L_0, since if the second option never pertains at any step n, we ensure that the interval $(0, 1)$ has type η and L_0 is wrong as it has an ω^* sequence of 4-blocks.

For L_1 we work similarly except now we try to find the 2 least 4-blocks and work between them. The extension to L_e is obvious and we leave the remaining details to the reader. $\square$

The key gap left open from the Fellner-Lerman-Rosenstein results is whether Theorem 5.20 has a converse; that is, can Theorem 5.19 be improved from Δ_3 to Π_2? This question was noted in several places such as Rosenstein [195], Fellner [62], Lerman-Rosenstein [132].

6 Effective Content of Some Classical Theorems: Subsequences, Embeddings and Automorphisms

In this section, we shall pursue one of the themes we mentioned earlier: investigating the effective content of classical theorems. Here the classical theorem (or fact) might say: every (*) ordering of type A has a (*) subordering of type B. The idea is to replace the (*) by the word 'computable' and see if the result remains true. If the result is now false we might then see what we can salvage.

We begin by looking at suborderings. The earliest result here is due to Tannenbaum (see Rosenstein [195]) and independently Denisov (see Goncharov and Nurtazin [81]). It is easy to see that each infinite linear ordering has an infinite subordering of order type either ω or ω^*. In contrast, we have

Theorem 6.1 (Tennenbaum, Denisov) *There is a computable linear ordering of order type $\omega + \omega^*$ with no infinite computably enumerable suborderings of order type ω or ω^*.*

Proof The proof is by a finite injury priority argument. Since we have already looked at this technique in some detail, it will suffice to describe the strategies for single requirements. We build $A = \cup_s A_s$ in stages to satisfy the requirements.

N_e a_e either has finitely many predecessors, or finitely many successors.

R_e If W_e is infinite, then it is not an ω sequence.

R_e^* If W_e is infinite, it is not an ω^* sequence.

To meet the requirement N_e, at some stage $s = s(e)$ we need to declare some finite sequence $a_{i_1}, \ldots, a_{i_k}$ such that either $a_{i_1}, \ldots, a_{i_k}, a_e$ is an initial segment of A, or $a_e, a_{i_1}, \ldots, a_{i_k}$ is a final segment of A.

The basic module for R_e is this. We wait till we see some *unrestrained* a_i occur in $W_{e,s}$ (for $i > e$). (Note that this is a finite injury argument and we ensure that the N_j, R_j and R_j^* for $j < e$ restrain only finitely many elements: if a_i fails to occur then $|W_e| < \infty$ so we win.) When $a_{i,s}$ occurs, we declare a_i and all of its current successors to be a final segment of A with priority e. For instance, we might have

$$a_0 a_1 \cdots a_j a_i a_{i_1} \cdots a_{i_k} = A_s.$$

Then $a_i a_{i_1} \cdots a_{i_k}$ is declared as restrained with priority e so that, with priority e we add no new elements to this set. Note that if we succeed with this restraint then W_e cannot be an infinite ω sequence as it contains a_i. The net effect is that a_{s+1} would be placed on the board between a_j and a_i as would subsequent a_t until some other R_k (R_k^*) acts. The R_e^* work dually and the N_e simply assert 'whatever a_e's current affiliation is (i.e. final or initial segment) preserve this'. The details fit together using a standard application of the finite injury argument. $\square$

It is not difficult to modify the above to show that A can be constructed so that both initial segments and final segments of A are hyperimmune. Recall that B is called hyperimmune if there is no infinite disjoint collection of canonical finite sets $\{D_{g(x)} : x \in \omega\}$ such that for all x, $D_{g(x)} \cap B \neq \emptyset$. Recall that a set D is called *cohesive* if there is no r.e. set Q with

$|D \cap Q| = |D \cap \overline{Q}| = \infty$. It is also not difficult to show that we cannot improve Theorem 6.1 to make the ω or ω^* pieces cohesive (see Soare [213, ch. X, Exercise 3.12]. In fact we cannot make the ω or ω^* pieces hyperimmune (Martin, see Jockusch[cor 4.5] [96]). Here B is called hyperimmune if there is no disjoint r.e. collection of r.e. sets (a *weak array*) $\{W_{f(x)} : x \in \omega\}$ such that for all x, $W_{f(x)} \cap B \neq \emptyset$.

The Tennenbaum-Denisov theorem has been generalised in many ways. One of the earliest was the original motivation for Watnick's theorem, which we met in Section 4. Recall that there we said that Watnick showed that τ is a Π_2 order type iff $\zeta\tau$ is computably presentable. Watnick proved the following extension.

Theorem 6.2 (Watnick [225]) *If τ is any Π_2 order type, there is a computable linear ordering A of type $\omega + \zeta\omega + \omega^*$ with no computable ω or ω^* sequences.*

Proof The idea is to take the proof of Section 4 and add a further layer of requirements

R_e W_e is not an ω sequence
R_e* W_e is not an ω^* sequence.

The details are a little complex, so we will only sketch how to combine the R_e's above with the Section 4 construction as meeting the requirements

M_e if $a_e \in \tau$ put an ω^* and ω sequence around a_e.

These in turn can be split into

$M_{e,i}$ if $a_e \in \tau$ put at least i immediate predecessors and i immediate successors around a_e.

The idea is as follows. To meet (e.g.) R_e we try to force W_e to contain a member of the final ω^* sequence. During the construction around (apparent) points of τ we will be building partial blocks of points. For the sake of R_e, if we see some x occur in a block of a_j for $j > e$ [2] then we can abandon totally this version of a_j (by priorities) and if, for instance, there is no a_i for $i < e$ between the a_j and the final ω^* region, simply add the a_j block (at s) to the ω^* block and pick a new a_j point around which to build a ζ block. See Figure 1 below for a typical scenario

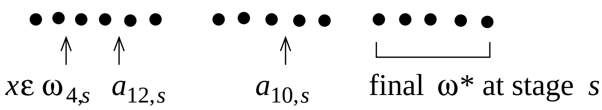


Figure 1:

Here we see $x \in W_{4,s}$ and only the apparent positions of $a_{12,s}$ and $a_{10,s}$ between x and the ω^* portion. We'd then get, at stage $s+1$, the situation of Figure 2.

The problems all stem from the fact that we could not use the idea above if we had $a_{3,s}$ instead of $a_{12,s}$ or $a_{10,s}$ as these have higher priority than R_4, so R_4 ought not to be able to erase them. The idea is then as follows. Suppose, for instance, we had $a_{3,s}$ in place of $a_{10,s}$. When we see

[2]Naturally, in the Π_2 version we would look for x in the block a_μ with μ of low priority than σ with σ the guess of R_e.

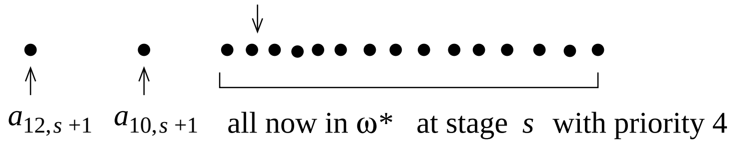


Figure 2:

x in $W_{4,\,s}$ we'd do the best we can: we'd add the $a_{12,\,s}$ block (including x) to the $a_{3,\,s}$ block to get the situation of Figure 3 at stage $s+1$.

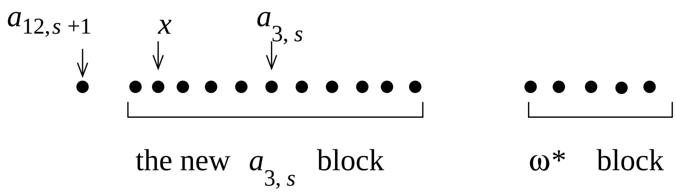


Figure 3:

Note that since x is to the left of $a_{3,\,s}$ and part of its block, if W_e contains an ω sequence, infinitely many $x' > a_{3,\,s}$ must occur in $a_{3,\,s}$'s block. *This is the key to the whole construction;* we wait till some x' occurs in $W_{e,\,s}$ and $a_{3,\,s}$'s block *more* than (say) $3+4=7$ further on than $a_{3,\,s}$ as in Figure 4

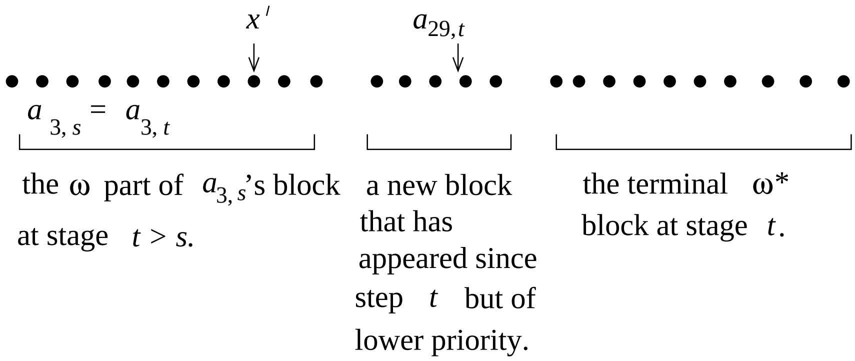


Figure 4:

We would then add all the points from x' onwards to the final ω^* block at stage $t+1$ with priority 4.

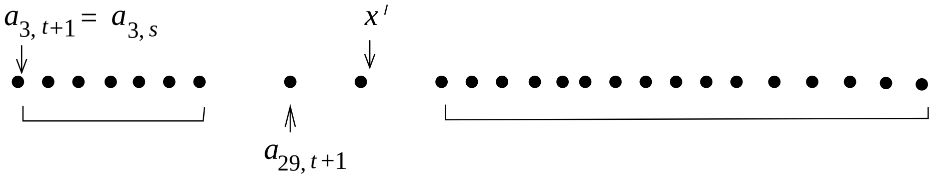


Figure 5:

The whole point is that such actions can only affect the ρ block around $a_{3,\,s}$ further and further out (i.e. of lower and lower priority) and so only finitely often injure some $M_{e,\,i}$.

The remaining details are rather messy, but one can fit all of the above together on a tree and get the desired results. □

Another generalisation of Theorem 6.1 goes back to the thesis of Hird [90] which appeared in [**?**]. One can view the Tennenbaum-Denisov theorem as constructing something bad isomorphic to $\omega + \omega^*$. We note that $\omega + \omega^*$ is a convex linear ordering.

Theorem 6.3 (Hird [90]) *Let A be a computable linear ordering with end points with P a co-re. subset. Then the following are equivalent.*

(i) There is a computable linear ordering $B \cong A$ with the image of P not r.e.

(ii) The subset P does not have both infimum and supremum in A.

Proof Actually we prove a slight extension of Hird's result since he assumed that P was co-re. in A. First if $(a, b) = P$ then P is clearly r.e. Furthermore, as P is co-re., if P is r.e. then P is computable. So to complete the proof, suppose that A has a convex computable subset P with no supremum (say). We need to build $f : B \cong A$ with $f^{-1}(P)$ not r.e.

The basic idea is fairly familiar. We have a partial isomorphism $f_s : \{b_0, \ldots, b_{n(s)}\} \to \{a_0, \ldots, a_{n(s)}\}$. We wait till we see some $b_i \in W_{e,s}$ with $f(b_i) \in P_s$. We then wish to remove b_i from $f^{-1}(P_s)$. This is okay since P_s has no supremum. We need only redefine the isomorphism so that it takes b_i (and all of its successors at stage s) to elements out of P (which we can do as P is computable and as A has end points) so that there are infinitely many elements $> P$ in A.

To make the argument valid we add priorities. That is, we don't allow our action for W_e to affect the $f_s(b_0), \ldots, f_s(b_e)$. □

One can apply the theorem above to the set $P = \{x : x \text{ has finitely many predecessors}\}$ in $\omega + \omega^*$ to make it not co-re. Indeed it is again easy to make it *immune* of any r.e. degree of unsolvability (this gives the Tennenbaum-Denisov theorem) or similarly to show

Theorem 6.4 (Hird [90]) *Let P be a co-r.e. convex subset of A. There is a computable $B \cong A$ via f with $f^{-1}(P)$ immune iff P satisfies one of the following:*

(i) P has order type ω and no supremum in A.

(ii) P has order type ω^* and no infimum in A.

(iii) P has order type $\omega^* + \omega$ and neither infimum nor supremum in A.

We remark that Hird obtained similar results for P being hyperimmune and extended Martin's result to show that

Theorem 6.5 (Hird [90]) *No convex subset of a computable linear ordering can be hyperimmune.*

Proof If P does not have order type ω, ω^* or $\omega + \omega^*$ and P is convex, then P cannot be (hyper)immune. Then we need only show the theorem for P of order type ω, ω^* or $\omega^* + \omega$. So, for instance, choose P of type ω. We build a weak array $\{C_e : e \in \omega\}$ so that for all e we meet

$$R_e : C_e \cap P \neq \emptyset.$$

Let p be the least element of P and $b_0, b_1, \ldots$ list $\{x : x \in A \text{ and } x \geq P\}$. At stage s choose the least e such that there is no element to the left of b_{s+1} in $C_{e,s}$ and add b_{s+1} to $C_{e,s}$. To see that all the R_e are met assume for an induction that this is so for all $i < e$ and additionally assume that each C_i is finite. Let s_0 be a stage where $C_{i,s_0} = C_i$ for all $i < e$. Let $s_1 > s_0$ be the first stage where $b_{s_1} \in P$. Then either b_{s_1} is added to C_{e,s_1} or there is already some $b_j < b_{s_1}$ in C_{e,s_1}. It is clear that as there are only finitely many b_k with $P \leq b_k \leq b_j(b_{s_1})$ that $|C_e| \leq |C_{e,s_1}| + b_{s_1}$. □

We remark that these results were only a small part of Hird's thesis which was devoted to analysing intrinsic properties of the lattice of relations of a computable structure under weak decidability conditions. Many known algebraic theorems can be deduced from Hird's model theoretic work.

Going back to the original observation about ω and ω^* sequences, we can salvage the following effective version.

Theorem 6.6 (Rosenstein [195]) *If A is a computable linear ordering then A has a recursive subordering of type ω, ω^*, $\omega + \omega^*$, or $\omega + \zeta\eta + \omega^*$.*

Proof If A has no ω nor ω^* computable sequence it must have a first and a last element. With the exception of these, each element must have a successor and a predecessor. Thus the order type of A must be $\omega + \zeta\alpha + \omega^*$ for some α. If $\alpha \neq \eta$, it is easy to see that A has a computable subordering of type $\omega = \omega^*$, giving the theorem. □

Rosenstein [195] asked if $\omega + \zeta\alpha + \omega^*$ was necessary in Theorem 6.6. Lerman answered this question affirmatively.

Theorem 6.7 (Lerman [130]) *There is a computable linear ordering with no recursive subordering of type ω, ω^* or $\omega + \omega^*$.*

Proof The argument is a straightforward Π_2 argument rather along the lines of a maximal set construction. We will confine ourselves to the technique used to meet two requirements, and leave the inductive strategies to the reader. We meet the requirement

R_e Either W_e is infinite, or W_e contains an element with infinitely many predecessors and successors (in W_e).

To meet R_0 we specify that a_0 is the least element of A, a_1 the greatest, and work in (a_0, a_1). R_0 does nothing until some $x \in W_{e,s}$ with $x \neq a_0, a_1$, so that $x \in (a_0, a_1)$. R_0 now freezes the interval (x, a_1) and specifies that the ordering is built in (a_0, x) until some y occurs in W_e with $y_0 \in (a_0, x)$. R_e will then switch to build the ordering in (x, a_1) until some z_0 occurs in (x, a_1) and then R_e switches back to (a_0, x) etc. Then if R_e acts infinitely often x will have infinitely many elements in W_e on each side. So when R_0 says 'work in (a_0, x)', R_1 can only pick some

$z \in W_{1,s}$ which occurs in (a_0, x). If R_1 switches to (x, a_1) we leave this version of R_1 on hold and start R_1 again in (x, a_1). If R_0 switches back, we can return to the old version of R_1. So if W_0 is infinite, R_0 imposes essentially no restraint on R_1. If W_0 is finite then R_0 imposes only finitely much restraint on R_1.

The inductive strategies for $n > 2$ requirements contain no surprises and the details follow as usual. $\square$

It is also possible to give an effective version different from Theorem 6.6.

Theorem 6.8 (Manaster) *If A is an infinite computable linear ordering then A has a Π_1 subset of type ω or ω^*.*

Proof Assuming A has no infinite computable ω or ω^* sequence, it has a least element a_0. We will define collections $y(i, s)$ and $z(i, j)$. We let $y(0, s) = a_0$ for all s. At stage 1 we let $y(1, 1) = a_1 = z(1, 1)$. We keep $y(1, s) = a_1$ until a stage s_1 occurs (it may not) such that there exists x with $a_0 < x < a_1 = y(1, s_1)$. We then define $y(1, s_1+1) = x$ and $z(1, 2)$. Now continue in the obvious way. Thus either for some s_n, for all $s > s_n$ we have $y(1, s) = y(1, s_n)$, or for all s there is a $t > s$ with $y(1, t) < y(1, s)$. In the latter case, $\{z(1, i) : i \in \omega\}$ is a computable ω^* sequence.

The inductive strategies are similar. We want to define $y(2) = \lim_s y(2, s)$ (if it exists). At each stage s we assume that the correct position of $y(1, s)$ is final. We must follow only two rules; we must always work right of $y(1, s)$ and never let $y(2, s) = y(1, t)$ for any $t \in \omega$. Note that in general, either there is an i with $\lim_s y(i, s)$ failing to exist (so an ω^* sequence) or, for all i there is an s_i with $y(i, s) = y(i, s_i)$ for all $s > s_i$ and so a Π_1 ω sequence consisting of $\{y(i) : i \in \omega\}$. $\square$

Actually the *proof* above shows that

Corollary 6.9 (Downey) *If A is in computable linear ordering then if A does not have a Π_1 ω (resp. ω^*) sequence, it has a computable ω^* (resp. ω) sequence. In particular, if A contains no computable ω nor ω^* sequences then it has both a Π_1 ω and a Π_1 ω^* sequence.*

One can well ask what can be said of the *individual* suborderings in the ordering L. For instance, suppose that the ordering has an infinite ω subordering. How hard can can it be to find such an ordering? The answer to this question is, perhaps, slightly surprising. We need a definition. The arithemtical hierarchy can be extended by allowing set quantifiers.

Let R be a relation and $S \subseteq \mathbb{N}$. The we can say that $R(S)$ holds iff for all initial segments $\overline{x}$ of S we have that $R(\overline{x})$ holds. Then we define a set A to be Σ_1^1 iff there is a arithmetical R such that

$$x \in A \text{ iff } (\forall S)R(x, S).$$

We say that A is Π_1^1 iff $\overline{A}$ is Σ_1^1, and extend to Π_n^1, Δ_n^1 and Σ_n^1 in the obvious way. We say that a set is *hyperarithmetical* if it is Δ_1^1. Kleene was the first person to study the such sets. Note that Π_1^1 sets are immensely complicated and can, in particular, compute all the n-th jumps of $\mathbf{0}$. Of interest to us is the following.

Theorem 6.10 (Kleene, see, e.g. Sacks [?]) *There is a computable linear ordering with an infinite decending sequence, but no hyperarithmetical such sequence.*

Proof sketch First one needs the classical result that there is a computable tree T of strings of integers with an infinite branch yet no hyperarithmetical branch. (See e.g. Rogers [194]) T can be linearized by the Kleene-Brouwer ordering ordering $\leq_K$ as follows. For strings σ we have $\sigma\widehat{\ }u <_K \sigma\widehat{\ }v$ if $v < u$ (note the inversion). We extend this to the whole tree by saying that $\sigma <_K \tau$ if either $\sigma \subseteq \tau$ or failing that, there exist $\sigma' \subseteq \sigma$, $\tau' \subseteq \tau$ with σ' and τ' both of the same length and $\sigma' <_K \tau'$.

Note that each infinite branch of T defines an infinite descending sequence in the ordering L. Conversely, suppose that $x_0 > x_1 > ...$ is a hyperarithmetical infinite descending sequence. Since and set of numbers of bounded length are clearly well ordered by $<_K$, we may as well suppose that the sequence $x_0, x_1, ...$ has uniformly increasing length. It is not difficult to prove by induction on length that for all n, there is a j such that for all $i \geq j$, $x_j \subseteq x_i$. One can then inductively define a hyperarithmetical branch. $\square$

Actually, the above also leads us to a nice result about initial segments of linear orderings. Suppose that $A + B$ is a computable linear ordering. What does that say about the complexity of the initial segment A? For instance, must A have a computable copy? The answer is no in a very strong way. (We here assume that the reader is familiar with the recursive ordinals.)

Theorem 6.11 *Let R be a computable ordering with infinite descending sequence, yet no hyperarithmetical such sequence.*

(i) **(Gandy)** *Let Q be the maximum well ordered initial segement of R. The Q is $\Pi^1_1 - \Sigma^1_1$, and has order type ω_1^{CK}, the first nonrecursive ordinal.*

*(ii)***(Harrison)** *R has ordertype $\omega_1^{CK}(1 + \eta) + \gamma$ where γ is a recursive ordinal.*

We omit the proof since it would take us a little far, but refer the reader to, e.g. Sacks [**?**], page 56 for the details.

Notice that the above leads to the slightly paradoxical observation that there are computable linear orderings $A + B$ such that the ordering $A + 1 + B$ has *no* computable presentation! (For to figure out if $x \in A$ one needs only compare with the point b representing the middle "1".

The above leads one to wonder if there are any conditions on A which guarantee computabe presentability. In his thesis [176] Mathew Raw studied this question and used a variation of the Feiner technique to construct a computable $A + B$ such that A is Π_3 and has no computable presentation. On the other hand he proved that if $A + B$ is computable and has computable adjacency relation, then if A is Π_1 the it has a computable copy. This result was extended as follows.

Theorem 6.12 (Ambos-Spies, Cooper, Lempp [?]) *If $A + B$ is computable, and A is Σ_2 then A is computably presentable.*

Porrf sketch The proof is not difficult but is a little delicate. First one proves that if I is a Σ_2 initial segment of L then there is a increasing Δ^0_2 sequence of elements of I cofinal in I. The proof is to use this sequence as boundaries for the Δ_2^0 isomorphism that is built. We build a computable $\langle M, < \rangle$. We use intervals (x_{n-1}, x_n) in I and map to (y_{n-1}, y_n) in M. This will be an isomorphism unless the intervals change. We will restart the map if possible unless we have already put more elements into the the (y_{n-1}, y_n) than (x_{n-1}, x_n), before they have settled down. Of course this is only a prblem if the intervals re finite. The idea is to first "adjust" the isomorphism slightly to the left of y_n. If all the intervals are finiet then the initial segment has type ω, which is fine. Otherwise, one is infinite and we can get to "catch up" the definition of the isomorphism. $\square$

Conversely, the following completely settles the exact quantifier level where non-presentability occurs.

Theorem 6.13 (Coles, Downey and Khoussainov [?]) *There is a Π_2 A with $A + B$ computable and such that A has no computable copy.*

Proof sketch The proof is a diagonalization but is rather more subtle than ones we have seen so far. The technique below promises more applications. Suppose that we wished to enumerate the blocks of a ordering, and in this case we suppose that the ordering is η-like in te sense that it has no infinite blocks. Suppose that we had $\emptyset'$ as an oracle. The we could determine successivies, and hence could demonstrate that if L is η-like, then $B(L)$ can be computed as the range of a $\emptyset'$ limitwise monotonic function. Here, a function f is *limitwise monotonic* if there exists a computable function $\phi(x, s)$ such that

(i) $\phi(x, s) \leq \phi(x, s + 1)$ for all $x, s \in \omega$,

(ii) $\lim_s \phi(x, s)$ exists for all $x \in \omega$,

(iii) $f(x) = \lim_s \phi(x, s)$.

The concept of a limitwise monotonic function is due to Khoussainov, Nies and Shore [**?**] and we have used the relativized notion. the crucial lemma is the following applied in relativized form.

Lemma 6.14 (Khoussainov, Nies and Shore) *There exists an infinite Δ_2^0 set S which is not the range of any limitwise monotonic function.*

Proof The proof of the Lemma is a very easy diagonalization argument which is an oracle construction. We must satisfy

$$R_e : S \notin \text{ra } \lim_s \phi_e(\cdot, s) \text{ or } \phi_e \text{ is not monotonic.}$$

The strategy for R_e is the following. We pick a witness x and put x into S. At each stage s we ask $\emptyset'$ if there is some $n \leq s$ such that $\varphi_e(n, s) = x$. If no such n exists then go to the next stage. Otherwise for each such n ask if there exists a stage $t > s$ such that $\varphi_e(n, s) \downarrow\neq x$. If the answer is yes, then x cannot be the pre-image of n by monotonicity. So we simply move to the next unconsidered n and if all such n are considered, simply go to the next stage. If the answer is no, then remove x from A, and cease all activity on R_e. Either $\varphi_e(x, t) \uparrow$ for some t, or $\varphi_e(x, t) = n$ for almost all t and $n \notin A$. $\square$

The proof of the theorem is complete via another coding argument.

Lemma 6.15 (Coles, Downey, and Khoussainov [?]) *If S is any Σ_3 set such that $m \in S \to m > 1$, then there is a computable $A + \omega^*$ such that A is η-like and $B(A) = S$.*

We will not prove Lemma in any detail as it is a relatively difficult infinite injury priority argument rather along the lines of a combination of Feiner's original argument and a Watnick type argument, and we have already met arguments along these lines.

Basically, let S be given via

$$x \in S \text{ iff } (existsn)(\forall y)(\exists z)R(x, n, y, z).$$

For each x as in Feiner's original theorem, we have to try to build a x block for each attempt at n. If we actually find a n such that the Π_2 condition $\forall y \exists z R(x, n, y, z)$ is satisfied then we build the x block. Otherwise we will kill the x block by either densifying it or have it incorporated in the final ω^* piece. For $x = 2$ and $n = 0$, we would begin with a 2-block $a_1 a_2$. While it looks bad, we regard it as part of the (finite segement built so far of the) ω^* block at the right end. We would be performing the rest of the construction to the left of this block. It at some stage, the Π_2 condition for $(2, 0)$ becomes verified another time, we would densify the work done (i. e. to the left of the 2-block) in the intervening stages, (wrong guess) and begin work to the right of the 2-block, building a partial densification between thi and the next block. Note that if this condition is infinitely often verified, then the material to the left will be dense, and other blocks will be built right of it. On the other hand if it is only finitely often verified, so either 0 is wrond or $x \notin S$, then the x will be incorporated in the ω^* piece. The inductive strategies work similarly except that we cannot densify things of higher priority than us. (The priority is given by putting all the pairs on a strategy tree.) The remaining details look like a Watnick style argument, and we refer the reader to the original paper for them.

To complete the proof, we can apply Lemma to an S which is the range of no $\emptyset'$ limitwise monotic function. Then A cannot be computably presentable as it is η-like. $\square$

A nice generalisation of the observation that every linear ordering has a subsequence from $\{\omega, \omega^*\}$is due to Dushnik and Miller [55] who showed that an infinite countable linear ordering has a nontrivial self-embedding. (That is an isotone $f : A \to A$ such that f is not the identity.) The proof of this result runs as follows. If A has an interval of order type ω (or ω^*) then define f to be the identity outside of this interval and map i to $i + 1$ in the ω interval. If A fails to have such an interval then A embeds $\mathbb{Q}$. Now we can use Cantor's proof that all countable linear orderings embed into $\mathbb{Q}$ to do the rest.

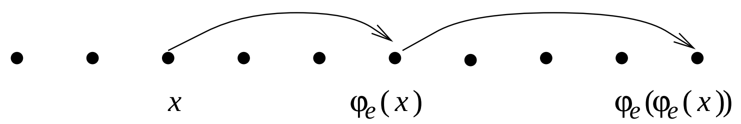
The Dushnik-Miller result is not effective:

Theorem 6.16 (Hay and Rosenstein (in Rosenstein [195])) *There is a computable linear ordering of type ω with no nontrivial computable self-embedding.*

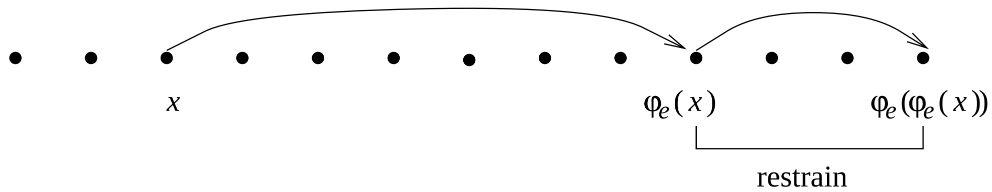
Proof sketch The proof is a finite injury priority argument. We build $A = \cup_s A_s$ so that we meet

N_e $\quad a_e$ has $< \infty$ many predecessors
R_e $\quad \varphi_e$ is not a nontrivial computable self-embedding.

To meet a single R_e, we wait until we see for some (unrestrained) x that $x \neq \varphi_{e,s}(x) \downarrow$ and $\varphi_{e,s}(\varphi_{e,s}(x)) \downarrow$. Note that if φ_e is to be a self-embedding, as we meet all the N_e, we need only worry if $x < \varphi_e(x) < \varphi_e(\varphi_e(x))$ $(< \cdots < \varphi_e^n(x)$ for all $n)$. The idea is simple. We simply add $|[\varphi_e(x),\ \varphi_e(\varphi_e(x))]| + 1 = n$ many new points between x and $\varphi_e(x)$ and then restrain $[\varphi_e(x),\ \varphi_e(\varphi_e(x))]$ with priority e. A typical situation is described below.



becomes



Then as $|[x,\ \varphi_e(x)]| > |[\varphi_e(x),\ \varphi_e(\varphi_e(x))]|$, φ_e cannot be isotone and 1-1. It is now routine to continue the obvious strategies in the finite injury manner to a set (6.16). $\square$

Actually, using a coding argument Downey and Lempp [**?**] have improved the above to read.

Theorem 6.17 (Downey and Lempp [?]) *There is a computable linear ordering L such that if f is a nontrivial self embedding of L then f can compute $\emptyset'$.*

A consequence of the Downey-Lempp Theorem is that the proof theoretical strength of the Dushnik-Miller theorem is ACA_0.

Of course some computable order types *always* have non-trivial self-embeddings. For instance

Observation 6.18 (Folklore) *If A is a dense computable linear ordering then A has a nontrivial computable automorphism.*

Proof Use Cantor's back and forth argument. $\square$

Observation 6.18 points out the most natural self-embedding: an automorphism. Extending earlier work of Rosenstein, Lerman and others, Schwartz was finally able to classify linear orderings with nontrivial computable automorphisms. Define a linear ordering to be *computably rigid* if it has no nontrivial computable automorphisms.

Theorem 6.19 (Schwartz [202, 203]) *If A is computably rigid then A has no interval of order type η. Furthermore if τ is an order type containing computable linear orderings, and τ has no interval of type η, then τ contains a computably rigid linear ordering.*

Proof Again we use a finite injury argument. Let A be an infinite computable linear ordering with no interval of type η. We construct $B \cong A$ via $f = \lim_s f_s$ so that we meet (for all $e \in \omega$),

R_e $\quad \varphi_e$ is not a nontrivial automorphism of B.

To meet R_e we ensure that either φ_e is not total, or ra $\varphi_e \neq \omega$ $(= A)$ or φ_e is not isotone and 1-1. Of course, as we go along, if we ever see $\cdots b_i < b_j$ with $\varphi_e(b_i) \not<_A \varphi_e(b_j)$ then we can stop as φ_e cannot be isotone. So we assume in the discussion below that this doesn't happen. The basic idea is simple and similar to Remmel's categoricity theorem.

Again suppose we knew the successivities of A. Then we'd wait till we saw some $b_1,\ b_2$ in B_s with $f_s(b_1) = a_1,\ f_s(b_2) = a_2$, and $\varphi_e(b_1) \neq b_1$ and $(a_1,\ a_2)$ successivity. So we can wait until $\varphi_e(b_2) \neq b_2$. Now, without loss of generality, we can suppose $\varphi_e(b_1) > b_1$ so that $\varphi_e(b_2) > b_2$. The basic idea is then to split $(\varphi_e(b_1),\ \varphi_e(b_2))$ by adding a new element b_{s+1} to the interval spoiling φ_e as an automorphism. To do this, as usual, we need to revise $f_{s+1}(\varphi_e(b_2))$, and so need to be careful – as in Remmel's theorem – that we don't move $f_s(b_0),\ \cdots,\ f_x(b_e)$. Also, as in there, we aren't able to identify the successivities in A, but know that each interval $[a,\ b]$ contains one. We simply guess then, as we did in Remmel's theorem, choosing the least one each time. The remaining details fit together in a familiar way via the finite injury method. □

Lerman and Rosenstein [132] raised the question of the classification of order types τ with a nontrivial Π_1 automorphism for each computable copy in τ. Kierstead [114] used an infinite injury argument to construct a computable linear ordering of type 2η with no nontrivial Π_1 automorphism. This was extended to any order type $\Sigma\{f(q)\ :\ q \in \mathbb{Q}\}$ with $f(q) \in \mathbb{N} - \{0\}$, with no interval of type η, and f a Π_2 function of Downey and Moses [52]. In view of Theorem 5.23 we thus have:

Theorem 6.20 (Downey) *If A is an η-like computable linear ordering with no interval of type η then there is a computable $B \cong A$ with nontrivial Π_1 automorphism.*

Proof By Theorem 5.23 if A is η-like then there is a Π_2 f with $\Sigma\{f(q)\ :\ q \in \mathbb{Q}\} \cong_{\text{rec}} A$. Now apply the Downey-Moses result. □

Most of the above were proved using (variations of) the idea of a choice set. A *choice set* for a linear ordering is a set of elements, one from each block. To prove Theorem 6.20 we used a variation of this idea. If A has only finite blocks, then a *strong choice set* is one where we have the first element from each block. Now if f is a nontrivial automorphism of L then $\{x,\ f(x),\ f^2(x),\ \ldots\}$ is an infinite ω or ω^* sequence in L. If L is computable and f is Π_1 this is a Σ_2 set. If g is a nontrivial Π_1 automorphism in B of Theorem 6.20 and x has no predecessors, nor do $g(x),\ g^2(x),\ \cdots$ and so $\{x,\ g(x),\ \cdots\}$ is an infinite Σ_2 subset of a strong choice set.

The Downey-Moses [52] argument is to show that B can be constructed with no strong choice set with an infinite Σ_2 subset. The original use of Σ_2 choice sets was due to Lerman and Rosenstein [132] who showed that *there is a computable linear ordering of type $\zeta\eta$ with no Σ_2 dense subset.*

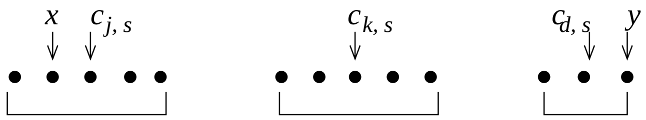
This argument was extended by Downey and Moses who showed

Theorem 6.21 (Downey and Moses [52]) *Every computable discrete linear ordering has a recursive copy with no Σ_2 choice set.*

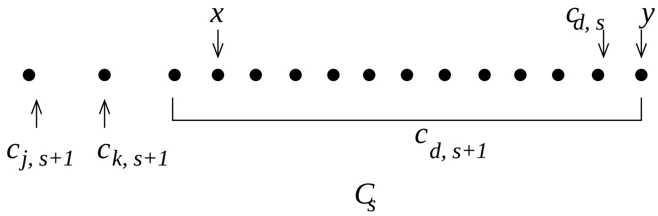
Proof sketch Again we use a variation of the Watnick construction. Let A have type $\zeta\tau$ so τ is a Π_2 order type. Let C have type τ with $C = \{c_0,\ c_1,\ \ldots\}$. We know from Section 5 how to produce $B \cong A$ via Watnick's construction. The extra layer of requirements this time are to diagonalise against all the Σ_2 sets. If $\{R_e : e \in \omega\}$ lists all the r.e. binary relations, we define T_e via

$$n \in T_e \text{ iff } \neg(\exists_x^\infty) R_e(n, x).$$

Then $\{T_e\}_{e\in\omega}$ enumerates all Σ_2 sets. In the Tennenbaum variation of the Watnick construction we defeated ω and ω^* sequences. Here we defeat T_e's by either making T_e too small or make $[x, y]$ finite for some $x, y \in T_e$. As in Theorem 6.2 there is no problem if x and y lie in (different) blocks with no blocks of higher priority between. As we did in Theorem 6.2 we might have



Then if $e < j, k, d$ we can amalgamate to get



Again we do have problems if $c_{j,s}$ and $c_{k,s}$ for instance had $j, k < e$. The point this time is that we will eventually get two we can combine if T_e is infinite. The fact that the T_e are Σ_2 and not computable creates some problems, but these can be overcome with the Π_2 method since we only need work with $\{e : |T_e| > e\}$ (which is a Π_2 set). So the outcome that we don't need action to meet T_e, is a Π_2 one and hence we can guess this on a tree of strategies. We don't give the details as they are quite tedious and contain no surprises. The reader is referred to Lerman and Rosenstein [132] or Downey and Moses [52]. □

We remark that Theorem 6.21 can be proven also with orderings of type $\omega\ \tau$ or $\omega^*\ \tau$. Kierstead [114] defined an automorphism to be *fairly trivial* if for all x, $|[x, f(x)]| < \infty$. If a nontrivial automorphism is not fairly trivial, Kierstead called it *strongly nontrivial.* As a corollary to Theorem 6.21 we have

Theorem 6.22 (Downey and Moses [52]) *Every computable discrete linear ordering has a recursive copy with no strongly nontrivial* Π_1 *self-embedding.*

Mathew Raw [176] independently proved the weaker result that there is a computable ordering of type $\zeta\eta$ with no strongly nontrivial Π_1 automorphism. He has also obtained the following related results:

(i) *There is a computable linear ordering A (with uncountably many automorphisms) such that A has a* Π_1 *automorphism and which has no computable automorphism any other computable presentation of the ordering.*

(ii) *The above problem 1 (for countably many automorphism) is equivalent to: Is there a rigid* Π_1 *initial segment of a computable linear ordering which is not isomorphic to a computable linear ordering?*

The results of Downey and Moses on η-like linear orderings (Theorem 6.20) are obtained in a similar vein to the proof of Theorem 6.21, only there one tried to force an element to have an immediate predecessor to stop it being a Σ_2 subset of a *strong choice* set. Finally, we remark that Theorems 6.21 and 6.20 are quite sharp since Lerman and Rosenstein observed

Observation 6.23 *Every computable linear ordering A has a Π_2 choice set.*

Proof The Π_2 relation

$$(\forall x < a)(\forall y)(\exists z > y)[(x < z < a) \vee (a < z < y)]$$

defines a choice set. □

There are quite a number of conjectures regarding Π_1 automorphisms. We list some below:

Conjecture 6.24 *(i)* **(Kierstead)** *Every computable copy of type τ has a strongly nontrivial Π_1 automorphism iff τ has an interval of type η.*

(ii) **(Downey and Moses)** *Every computable copy of τ has a fairly trivial Π_1 automorphism iff τ has elements x, y with (x, y) consisting only of block of type ζ.*

(iii) **(Downey and Moses [52])** *Every computable copy of τ has a strongly nontrivial Π_1 self embedding iff τ has a strongly η-like interval (i.e. (x, y) such that (x, y) is infinite and only has block of size $\leq k$).*

(iv) **(Downey and Moses [52])** *Every computable copy of τ has a fairly trivial Π_1 self-embedding iff τ has an interval (x, y) consisting only of blocks of type ω or ζ or only of blocks of type ω^* or ζ.*

(v) **(Kierstead [114])** *If a computable linear ordering has neither intervals of type ζ nor η then it has a copy with no nontrivial Π_1-automorphism.*

Of course the basic question is to characterise the order types of Π_1-rigid computable linear orderings.

Returning now to computable self-embeddings, we have the following that was surely noticed by all workers in the area. It was probably first noted by Watnick and Lerman.

Theorem 6.25 *If A is a strongly η-like computable linear ordering, then A has a nontrivial computable self-embedding.*

Proof Let $A = \Sigma\{f(q) : q \in \mathbb{Q}\}$ with $f(q) \in \{1, \ldots, n\}$ for all q. Knowing n we will know that when we see $(n+1)n$ many elements, elements $1,\ n+1,\ 2n+1,\ \ldots$ must all belong to different blocks. This clearly allows us to define the image of any block via a self-embedding. □

Lerman, Watnick, Moses, and others have conjectured that Theorem 6.25 characterises the relevant order types.

Conjecture 6.26 (Folklore) *If τ is an ordertype containing computable linear orderings such that all computable linear orderings of type τ have a nontrivial recursive self-embedding, then τ has a strongly η-like interval.*

In support of Conjecture 6.26 it is possible to use the techniques of (e.g.) Theorem 6.21 to show that *if τ is η-like but has no strongly η-like intervals then a computable order type τ has a computable copy with no nontrivial recursive self-embedding.*

However, Conjecture 6.26 remains unresolved and, furthermore, the situation is quite complex as we will now see. In all of the constructions we have seen so far, counterexamples were obtained *uniformly* in the sense that there is a computable function f such that $W_{f(e)}$ is the relevant witness to W_e (if appropriate). For instance, in Remmel's theorem, Remmel actually shows that there is a computable function f such that if W_e is a computable linear ordering with infinitely many successivities, then $W_{f(e)}$ is isomorphic with W_e yet $W_{f(e)}$ is not computably isomorphic to W_e. It is unusual (but not unheard of) for constructions to be nonuniform. That is, for an example to exist, yet no f *selecting* an example to exist. The first example in classical recursion theory of this phenomenon was due to Lachlan [125]. We have seen it in the proof of our extension of Manaster's result where we don't know which happens i.e. either an ω or an ω^* sequence. We will now see that if Conjecture 6.26 is correct, there is no uniform procedure to prove it.

Theorem 6.27 (Downey) *There is no uniform procedure which, when given a computable linear ordering A produces a computable $B \cong A$ such that if A contains no strongly η-like interval then B has no nontrivial computable self-embedding.*

Proof We build an infinite $(A, \leq)$ such that A has no strongly η-like intervals and we meet the requirements

R_e If $W_e \cong A$ then $(W_e, \leq)$ ha a nontrivial computable self-embedding.

The result will then follow by the use of the recursion theorem.

To meet R_e, we shall define a mapping $g_e = g$. If we let $z(e, i) = z(i)$ denote the i-th member of W_e in order of appearance, we must meet for all i.

$R_{e,i}$ $g(z(i))$ is defined.

To achieve (i) we make sure that either A is finite or, for almost all j, a_j (= the jth members of A) is part of an ω or ω^* or $\omega + \omega^*$ sequence. To do this we define $B(a_j, s)$ to denote the block containing a_j at stage s. We will ensure that $B(a_j, s) \subseteq B(a_j, s+1)$ and hence we meet for almost all j

$$P_j \; : \; \lim_s |B(a_j, s)| = \infty.$$

To meet the requirements above, we proceed as follows.

We begin by enumerating a_0. We win unless W_e enumerates $z(0)$. (If not $|A| = 1$ and $|W_e| = 0$). It is now our goal to find a j with $z(j) < z(0)$, so to meet $R_{e,0}$ and define $g(z(0)) = z(j)$. To do this, we define $a_1 > a_0$. Now W_e can only respond in one of 3 ways. It can enumerate

$z(1) < z(0)$, it can enumerate $z(1) > z(0)$ or it can do nothing. In the last case we win, in the second case we meet $R_{e,0}$. Only the first case creates any problems.

The idea is that we continue in this way (i.e. put $a_2 > a_1$, etc.) until some $z(j) > z(0)$ occurs. Note that if no such $z(j)$ occurs, we win as then $W_e \ncong A$. The reason is that A is an ω-sequence yet W_e contains $z(0)$, a point with infinitely many points to the left of it.

Note that for the above strategy $|W_{e,s}| = |A_s|$ or $|W_{e,s}| = |A_s| - 1$ for all s. Of course, we control the enumeration of W_e to keep $|W_{e,s}| \leq |A_s|$ since if $|W_{e,s}| > |A_s|$ we can stop $A_s = A$ and force $W_e \ncong A$. A typical situation is given below in Figure 6:

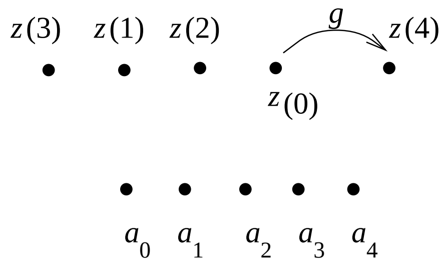


Figure 6:

At the end of this process (i.e. when we meet $R_{1,0}$) we then begin a 'clean up' process that helps to meet the P_j as we will see. The idea is simple. Before we meet any other $R_{e,i}$ we will ask that each *current* point of A is a member of block of at least one more member (i.e. if this happens at stage s). Thus we immediately give each member of A s successors. For example if the s_1 function of Figure 6 happened at stage 10 we'd have (at stage 11) the situation of Figure 7 below:

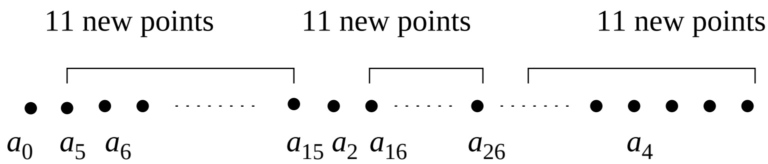


Figure 7:

The reason for this becomes clear later. Now we don't do anything until $|W_{e,t}| = |A_t|$ again at some $t > s$.

The problems that can occur are that we need to force points between others. A typical bad situation is described below:

This situation in Figure 8 is not quite accurate as there would always be more points in the actual construction.

We must try to get a point $z(j)$ with $z(0) < z(j) < z(1)$. *This is the heart of the construction.*

The basic action is to put a new point a_6 between a_3 and a_4. W_e can respond in one of 4 ways:

(i) no response (global win)

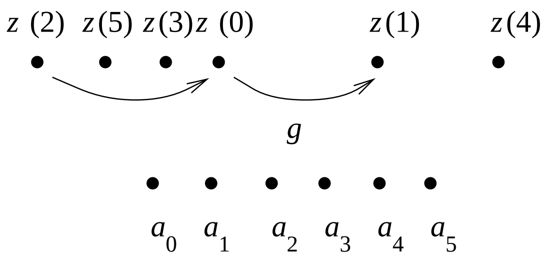


Figure 8:

(ii) $z(6) < z(0)$

(iii) $z(6) > z(0)$

(iv) $z(0) < z(6) < z(1)$ (win $R_{e,3}$).

In case (ii) we'd put a point between a_6 and a_4 and in (iii) we'd put a_7 between a_6 and a_3. Continuing, assuming (i) and (iv) do not occur, there are three outcomes given by the order type of A we get

Outcomes 6.28

(i) $\omega + k$: (iii) happens only finitely often and (ii) infinitely often.

(ii) $k + \omega^*$ (iii) happens infinitely often and (ii) only finitely often.

(iii) $\omega + \omega^*$ both (ii) and (iii) happen infinitely often.

In case (i) we win since the order type of W_e must be of the form $B + (k + 1)$; in (ii) it is $(k+1)+B$ and in (iii) W_e must have $z(0)$ and $z(1)$, two points with infinitely many points each side of them. In any case $W_e \not\cong A$.

Of course if (ii) always occurs, then we get the desired self-embedding and A has the good order type in the sense that all elements are members if infinite blocks. The reason, of course, is that after the clean-up action, the place we need to add points to force W_e to respond can always be between two of the *added block points* and so achieve the goal of adding one point to each block. $\square$

The technique above can be applied to Conjecture 6.24. There seems no clear way to combine R_e strategies in the above for all R_e. One possible way is to use a $\mathbf{0}^{(3)}$ or $\mathbf{0}^{(4)}$ version of the 'separator' technique of Section 8, but at present this author can see no way to get this to work (and has tried very hard to do so).

One extension of Theorem 6.27 can be obtained if we control the complexity of the isomorphisms.

Theorem 6.29 (Downey) *There is a computable linear ordering A with no η-like interval such that for all computable B isomorphic to Z* **via** *a* Δ_2 **isomorphism**, *B has a nontrivial computable self-embedding.*

Proof We meet the following modified R_e:

R_e If $W_e \cong A$ via $\varphi_e(\ ,\ )$ with φ_e Δ_2 then W_e has a nontrivial computable self-embedding.

Here $(W_e,\ \varphi_e)_{e\in\omega}$ is an enumeration of all r.e. subsets of $\mathbb{Q}$ and binary partial computable functions, so to meet $\hat{R}_e$ we need to ensure that either for some x, $\lim_s \varphi_e(x,\ s)$ does not exist, or $\varphi_e(x,\ s)$ is not total, or φ_e is not a homomorphism or W_e has a nontrivial computable self-embedding.

Using an idea along the lines of 'separators' we meet in Section 8, we divide A into infinitely many intervals $\{I_e\}_{e\in\omega}$. We play R_e on interval $I_e = [a_e,\ b_e]$. On this interval, we plan to perform the Theorem 6.27 strategy.

The idea is simple. We replace A of Theorem 6.27 by I_e and replace W_e by $W_e \cap [\varphi_e(a_e),\ \varphi_e(b_e)]$. So at stage s, we replace this by $\{z :\ z \in W_{e,s} \text{ and } \varphi_{e,s}(a_{e,t}) \leq z \leq \varphi_{e,s}(b_{e,t})\}$. If we reach a stage where $\varphi_e(a_e,\ t_1) = \varphi_e(a_e)$ and similarly b_e for all $t > t_1$, then we can win *between* $\varphi_e(a_e)$ and $\varphi_e(b_e)$ in exactly the same way as we did in Theorem 6.27. If t_1 fails to exist, we need only ensure that $[a_e,\ b_e]$ is not η-like and we win as $\varphi_e(a_e)$ has no value, so φ_e does not represent a Δ_2 function. $\square$

We remark that Theorem 6.29 is of some technical interest since most isomorphisms we have met, or are constructed for such problems, are Δ_2. There are some examples of other ones. For instance in Section 8 we have an example where the witness is Π_2. It is not clear if the above can be pushed to Δ_3. Personally, I feel if it can be then there is an A such that all $B \cong A$ have nontrivial computable self-embedding, or there is a proof to the contrary where things are constructed at the Δ_3 level. One interesting possibility is that we could push Theorem 6.29 perhaps for all arithmetical isomorphisms.

7 Linear Extensions of Partial Orderings, Dilworth's Theorem, and Effective Dimension

In this section, we will focus upon dimension and extension theory for orderings. We will be concerned with computable *partial* orderings, where $(A, \leq)$ is a computable partial ordering if A is identified with ω and given $x, y \in A$ we can decide which of $x < y$, $y < x$, $x = y$ or $x \,|\, y$ hold. An *extension* P of poset R is one where P and R have the same domains, and such that $x \leq y$ in P means $x \leq y$ in R. If R is a linear ordering, then we say that R is a *linear extension* of P. An old result of Szpilrajn [214] says that *any poset has a linear extension.* This leads to a theory of *dimension* for posets. The dimension of a poset P is the smallest ordinal n such that P can be expressed as the intersection of n linear extensions. This idea has led to an interesting theory which contains such interesting theorems as Dilworth's decomposition theorem. We shall look at some effective aspects of this theory in this section.

It is not difficult to see the following

Observation 7.1 (Folklore) *If P is a computable partial ordering, then P has a computable extension.*

Proof The thing to note is that, with care, Szprilrajn's proof is effective. Let $P = \{a_0, a_1, \ldots\}$. Gödel number all the pairs from P as $n_0, n_1, \ldots$ At stage s, we attend to n_s. Let n_s be the Gödel number of (x, y). If $x \leq y$ or $y \leq x$ (in R_s) to go stage $s+1$. If $x \,|\, y$ declare $x \leq y$ (in R_{s+1}) and for all $a \leq x$ and $y \leq b$ (in R_s) declare $a \leq b$ (in R_{s+1}). Here $R_0 = P$ and we set $R = \cup_s R_s$.

Then R is the desired computable ordering. Note that R is computable as follows. Given m_1 and m_2 we wish to decide if $m_1 \leq m_2$ or $m_2 \leq m_1$. Let n_k be the Gödel number of (m_1, m_2). Then in R_{k+1} we will know which of $m_i \leq m_{i+1}$ holds since either we have $m_i \leq m_{i+1}$ in R_k or we will attend to (m_1, m_2) in R_{k+1}. $\square$

Much of the classical research about linear extensions has focused upon *preservation.* That is, if R has property Q does there exist an extension of R with property Q? (One could also look at the *Q-dimension* of R, namely the least n, if any, such that R can be written as the intersection of n orderings all with property Q. I am not aware of much work in this area.) For instance, we can show that if R lacks an ω^* sequence (i.e. it is well founded) then R has a well founded linear extension. (Bonnet [21]). In fact, Bonnet (see Bonnet & Pouzet [23]) completely classified the countable linear order types τ such that if R is countable and τ-free then R has a τ-free linear extension.

Question 7.2 *Is there an effective analogue of Bonnet's result?*

Rosenstein, Kierstead and Statman have investigated the effective content of the earlier result of Bonnet.

Theorem 7.3 (Rosenstein, Kierstead (see Rosenstein [196])) *If P is a well founded computable poset, then P has a well founded computable extension.*

Rosenstein [196] remarks that the standard proof fails to prove Theorem 7.3. The standard proof lets A_0 be the set of R-minimal elements and A_j to be the minimal elements of $A - \cup_{i<j} A_i$. Then one well orders the A_j's and puts the rest together. This proof would give a Π_2 extension.

Proof of Theorem 7.3 The idea is to modify the proof of Observation 7.1 (essentially). Suppose, at stage s, we have P an ordering of $\{a_0, \ldots, a_{s-1}\}$ extending P on $\{a_0, \ldots, a_{s-1}\}$. Suppose the P-ordering is $a_{1_0} < \cdots < a_{1_{s-1}}$. Find the R_s-smallest a_{i_t} such that $a_s <_P a_{i_t}$ and set $a_{i_{t-1}} < a_s < a_{i_t}$. If a_{i_t} does not exist then put $a_s > a_{s-1}$. Then P is a linear extension so we need only prove that P is well founded.

Suppose $\{a_{j_0} > a_{j_1} > \cdots\}$ is an ω^* sequence in R. By Ramsey's theorem, we can assume that for all $k \neq m$, $a_{j_k} \,|\, a_{j_m}$ in P since we know P is well founded. We can also suppose that $j_k < j_{k+1}$ for all k.

Since $a_{j_k} <_R a_{j_0}$ but $a_{j_k} \,|_P\, a_{j_0}$ it follows that there is some c_k with $a_{j_k} <_P c_k <_R a_{j_0}$. Note that as $a_{j_k} \not<_P a_{j_0}$, $\quad c_k \not<_P a_{j_0}$. It follows, that if $c_k = a_{i_t}$, then $i_t < j_0$, by construction. Choose i_t least. It follows that there is some *least* d_1 with $a_{j_i} <_P d_1 <_R a_{j_0}$ for infinitely many i and hence we can thin the sequence to get, without loss of generality, $a_{j_0} \;{}_R\!>d_1 \;{}_P\!>a_{j_i}$ for all i. We now continue inductively to get a sequence $a_{j_0} > a_{j_1} > \cdots$ with $a_{j_i} \;{}_R\!>d_i \;{}_P\!>a_{j_{i+1}}$. Using the minimality argument above it is easy to see that $d_1 \;{}_P\!>d_2 \;{}_P\!> \cdots$ giving a contradiction. $\square$

A couple of comments on this proof. First, in Rosenstein [196] it was noted that the argument gives a (new) proof that if *α is an infinite cardinal and $\alpha \not\leq |P|$ then there is a linear extension R of P such that $\alpha \not\leq |R|$.* Secondly, the argument is very reminiscent of the 'minimal bad sequence' arguments used in well quasi-ordering and better-quasi-ordering proofs. This suggests that there may be some connections between computability and bqo theory.

It was noted by Rosenstein [196] that a key obstacle in solving Question 7.2 is dealing with scattered types.

Question 7.4 (Rosenstein [196]) *If R is scattered and computable, does R have a scattered computable extension?*

The classical version is given in Bonnet and Pouzet [23] and Galvin and McKenzie (unpublished), and is part of Bonnet's classification.

We remark that it is not totally clear what the correct analogue for Question 7.2 ought to be. As if often the case, a classical theorem can have multiple effectivizations. For instance, we can ask if being *'computably well founded'* is preserved. Here we say that an ordering A is *computably well founded* (resp. *computably scattered*) if it has no *computable* ω^* sequence (resp. does not *computably embed* η).

Theorem 7.5 (Rosenstein and Statman (in Rosenstein [196])) *There is a computably well founded computable partial ordering with no computably well founded computable linear extension.*

Proof Let T be the computable collection of orderings σ such that $T = \cup_s T_s$ defined as follows:

$$\sigma \notin T_s \text{ iff } (\exists t)(\exists \gamma)(\gamma \in W_{e,t} \text{ and } t < s \text{ and } 1h(\gamma) > 2e \text{ and } \gamma \subseteq \sigma).$$

Then T is a computable tree and is infinite by Köning's Lemma. Define the usual partial ordering on T via λ, $\gamma > \sigma$ iff $\gamma \subseteq \sigma$. Then $>$ is a computably well founded partial ordering on T since if Q is a branch of T then $W_e \not\subseteq T$ for any e.

Let B be a computable linear extension of T. Define a computable ω^*-sequence Q as follows. At stage s, we choose b_s from level s. At stage 0 let $b_0 = \lambda$. At stage s let P_s denote the collection of immediate predecessors of $Q_{s-1} = \{b_0, \ldots, b_{s-1}\}$ met already in Q_{s-1}. Let a_s be the B-largest element of P_s. Then Q is clearly computable. The claim is that $a_{s+1} <_B a_s$ for all s. If $a_s <_B a_{s+1}$ then $a_{s+1} \notin Q_t$ for all $t < s$. But $a_{s+1} \in Q_s$, and hence a_{s+1} is an immediate predecessor of a_s, so $a_s <_A a_{s+1}$ and we are done. □

Question 7.6 (Rosenstein [196]) *What is the situation with computably scattered partial orderings?*

Rosenstein [195, 196] observed that one can resurrect effective versions using the arithmetical hierarchy.

Theorem 7.7 (Rosenstein [196]) *Let R be a computable partial ordering. Then*

(i) If R is computably well-founded, it has a Δ_2 computable well founded linear extension.

(ii) If R is computably scattered, it has a Δ_2 computably scattered linear extension.

Proof (eg (i)) One constructs an extension of R ensuring that if $W_e = \{a_0, a_1, \ldots\}$ is infinite then for some $i < j$ that $a_j <_B a_i$. Using a priority argument, this needs only a Δ_2 oracle. □

Question 7.8 *Is Δ_2 the best we can do here? Can it be replaced by Π_1?*

One can also ask in Question 7.8 what degrees such extensions might have. There are many other preservation properties of linear extensions for which one might ask the effective content (see Bonnet and Pouzet [23]). Another topic of interest here is

Question 7.9 *Classify the computable partial orderings with exactly one computable linear extension.*

Recently Slaman and Woodin [**?**] considered the question of finding necesary and sufficient conditions for a poset to have a dense linear extension without endpoints. (This was a question of Loś.) Slaman and Woodin demonstrated that the problem has no simple solution by proving.

Theorem 7.10 (Slaman and Woodin [?]) $\{e : W_e$ *is a computable partial odering of* ω *and there is a endpoint free dense linear extension of* $W_e\}$ *is* Σ^1_1 *complete.*

The proof of the above would take us beyond the scope of the present paper, and involves coding a new Π^1_1 set which Slaman and Woodin call "almost perfect well founded trees."

The next topic we turn to is the topic of dimension. The *computable dimension* of a poset A is the least d such that A can be *computably embedded* into $\mathbb{Q}^d$. This corresponds to the classical notion of dimension introduced by Dushnik and Miller [55] since all countable linear orderings are embeddable in $\mathbb{Q}$. As we might expect, the Dushnik-Miller theorem has a number of effectivizations, even for *two dimensional partial orderings* (2 dpo's). 2dpo's have been studied quite extensively (see eg Kelly and Trotter [103], Trotter and Moore [221], Baker, Fishburn and Roberts [13]) and as the 'next simplest' poset beyond a linear ordering, their combinatorics and computable combinatorics are of some interest. Manaster and Rosenstein [141] initiated the study of computable 2dpo's. The notions of computable 2dpo and computable poset with computable dimension two are quite different as we will see.

Theorem 7.11 (Manaster and Rosenstein [141]) *There is a computable 2dpo that does not have computable dimension 2.*

In fact, Manaster and Rosenstein improved Theorem 7.11 to show

Theorem 7.12 (Manaster and Rosenstein [141]) *There is a computable 2dpo B that is not isomorphic to a computable subordering of* $\mathbb{Q}^2$.

Lemma 7.13 (Manaster and Rosenstein [141]) *There is a sequence of finite 2dpo's, none of which is embeddable in the other.*

Proof Let $n \geq 4$ and let D_n be the ordering of the $2n+1$ points $d_1, \ldots, d_n, q_1, \ldots, q_{n-1}, p_1, p_2$ ordered by the transitive closure of:

$$q_i < d_i \text{ and } q_i < d_{i+1} \text{ for } 1 \leq i < n$$

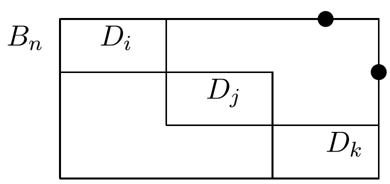
$$d_i < p_1 \text{ for } 1 \leq i < n \text{ and } d_j < p_2 \text{ for } 1 < j \leq n.$$

It is routine to show that if D_m is embeddable into D_n then $m = n$. $\square$

Proof (Theorem 7.12) The argument is quite elegant. The plane is divided into a sequence of disjoint boxes $B_1, B_2, \ldots$. Into B_n we put all of the points of D_{3n+i} for $i \in \{1, 2, 3\}$ where $\{D_j : j > 4\}$ lists a computable set of 2dpo's of Lemma 7.13. Use the even numbers to do the Gödelization. B_i may contain two additional points. The only new relations are determined by the construction and that if $i < j$ and $x \in B_i$ and $y \in B_j$ then $x <_B y$.

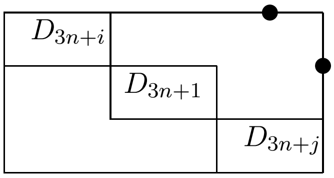
Let W and V be two computably enumerable *computably inseparable* sets. That is $W \cap V = \emptyset$ and there is no computable set R with $W \subset R$ and $R \cap V = \emptyset$. (These are easy to construct. At stages s, if $W_s \cap W_{e,s} = \emptyset$ and there is a $z \in W_{e,s}$, $z \notin V_s$ and $z > 4e$, put z into $W_{s+1} - W_s$ (similarity V_s) and let $W = \cup_s W_s$, $V = \cup_s V_s$).

The picture of B_n is

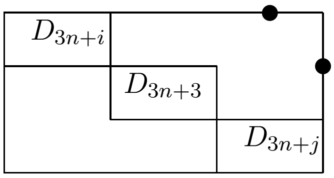


At stage s, if n occurs in W_s or V_s the first two odd numbers not used are placed in B_n as marked. In detail, if $n \in W_s$, the first point is placed above all points of D_{3n+1} and D_{3n+2} but made incompatible with D_{3n+3} and the second placed above all of D_{3n+1} and D_{3n+3} but incompatible with D_{3n+2}. If n occurs in V, the first of the two points is placed above all the points of D_{3n+2} and D_{3n+3}, but incomparible with D_{3n+1} whilst the second above D_{3n+1} and D_{3n+3} but incompatible with D_{3n+2}. Clearly R so construed is computable. Moreover it is a 2dpo via the representation $B_0^* < B_1^* < \cdots$ with B_n^* described as follows.

$B_n^* = D_{3n+i} < D_{3n+j} < D_{3n+k}$ if $n \notin W \cup V$, $i \neq j \neq k \in \{1, 2, 3\}$

$B_n^* =$ 

with $i \neq j$ and $i, j \in \{2, 3\}$
if $n \in W$

$B_n^* =$ 

with $i \neq j$ and $i, j \in \{1, 2\}$
if $n \in V$.

Suppose now that B is isomorphic to a computable subset A of the plane. We claim that we can computably separate W and V by a set R. Search B until subsets D'_{3n+1}, D'_{3n+2}, D'_{3n+3} are found with $D'_{3n+i} \cong D_{3n+i}$. These clearly must exist. Moreover, by the construction of B, we know that they must be in the orientation given by the B_n^*'s. Let P_i be a point of D'_{3n+i}. Determine which is left uppermost and which is right lowermost, and let P_t be the last one. If $t = 1$ put $n \in R$ otherwise put $n \notin R$. R is then a computable separator of W and V. $\square$

A similar argument will show

Theorem 7.14 (Manaster and Rosenstein [141]) *There is an computably enumerable subordering of $\mathbb{Q}^2$ not isomorphic to a computable one.*

(This should be compared with the corresponding situation for linear orderings.) While $\mathbb{Q}^2$ is not the universal object for 2dpo's, Manaster and Rosenstein did manage to find one.

Theorem 7.15 (Manaster and Rosenstein [141]) *There is a computable 2dpo, U such that if R is computable 2dpo then R is isomorphic to a computable subset of U.*

Proof sketch Use an infinite set of boxes B_i as before, now each containing an infinite computable set $R_e = \{b_0^e, b_1^e, \ldots\}$. We split B_e into two portions, one for 'padding' and one to embed L_e, the e-th 2dpo (which may be finite). We declare all the points in the lower portion to be less than those in the upper. At stage s we put b_e^s as part of an ω-sequence in the lower portion if no new element occurs in $L_{e,s}$. If a new element occurs, put it in the appropriate place in the upper portion to keep it isomorphic with $L_{e,s}$. $\square$

Manaster and Remmel [138] made similar investigations into *dense* computable 2dpo's. Since our concern is mainly with linear orderings we shall be rather brief here. By a dense 2dpo we mean one that is isomorphic to a topologically dense subset of $\mathbb{Q}^2$. Some special dense 2dpo's are D_2, a computable subset of $\mathbb{Q}^2$ with no 2 points collinear (this characterises D_2 up to isomorphism (see Manaster and Remmel [139]) and $D_2^{m,n}$ where this is the result of augmenting D_2 by m horizontal and n vertical lines (in some computable fashion). D_2 is like a universal structure for computable dense 2dpo's.

The density of a 2dpo adds quite a lot of room for manoeuvring. For instance, in contrast to Theorem 7.12 we have via a Cantor back-and-forth argument:

Theorem 7.16 (Manaster and Rosenstein [141]) *The following are equivalent for a dense 2dpo P*

(i) P is computably isomorphic to a computably presented 2dpo.

(ii) P is computably isomorphic to a computable subset of D_2.

(iii) P is isomorphic to an computably enumerable subset of D_2.

Proof sketch Evidently (ii) $\Rightarrow$ (iii) $\Rightarrow$ (i), so we need only show that (i) $\Rightarrow$ (ii). In Manaster and Remmel [138] it is shown that the coordinate orderings $\leq_1, \leq_2$ for *dense* 2dpo's are definable in the theory of dense 2dpo's via constants i and j which are interpreted as incompatible, $i\,|_P\,j$ and agreeing $i <_1 j <_2 i$. So we can view $r <_1 s$ meaning r is left of s and $r <_2 s$ means r is below s. It is not necessary that $<_1$ and $<_2$ be computable just because $<$ is, but Manaster and Remmel show they are computably enumerable as follows. Define $x <_1^* y$ to mean $x <_1 y \vee (x =_1 y$ and $x <_2 y)$ and similarly $x <_2^* y$ to mean $x <_2 y \vee (x =_2 y$ and $x <_1 y)$. Then one can show that $<_1^*$ and $<_2^*$ are computable using the fact that $<$ is computable as $<_1^*$ and $<_2^*$ are both computable, we can define a computable f with $x <_k^* y$ iff $f(x) <_k f(y)$ in D_2 for $k = 1, 2$ via the density of D_2. This allows us to show that $\overline{f(P)}$ (in D_2) is computably enumerable and hence $f(P)$ is computable. $\square$

This proof also shows that a dense 2dpo P is isomorphic to a computable subset of D_2 iff P is isomorphic to an computably enumerable subset of D_2. The technique of proof would also be able to prove that D is isomorphic to a Π_n subset of D_2 iff it is isomorphic to a Σ_{n+1} subset of D_2 (as we saw with the linear orderings.)

It is not true that an computably enumerable P is computably isomorphic to a computable subset of D_2. Actually, here we need some care with what we mean by D_2. Whilst D_2 is characterised up to isomorphism, it is not so up to *computable* isomorphism. The key notion is that of a *decidable* presentation. Recall that for a computable model $\mathbf{A}$ we say that a presentation is decidable if there is a computable procedure R such that given any first order formula φ in the language of $\mathbf{A}$, R can decide if φ is true of $\mathbf{A}$ or not. Using a wait and see argument, Manaster and Remmel showed the following.

Theorem 7.17 (Manaster and Remmel [138]) *There is an computably enumerable presentation of D_2 not computably isomorphic to a computable subset of a decidable presentation of D_2.*

Proof sketch We build a computable subset D of $\mathbb{Q}_2$ that is isomorphic with D_2. Let B denote the decidable presentation of D_2. We meet for $e \in \omega$

R_e $\quad \varphi_e$ is not a computable embedding of D into B.

To aid the construction, one builds a Δ_2 isomorphic $f = \lim_s f_s$ from D to B, in a fashion similar to many of our earlier arguments.

The basic idea is essentially the following. If R_e is not yet (declared) satisfied, we wait till there exist a_e and b_e with

(i) $f^s(a_e) = \langle 2e,\ 2e+1\rangle$, $\quad f^s(b_e) = \langle 2e+1,\ 2e\rangle$,

(ii) $\varphi_e^s(a_e)\downarrow$ and $\varphi_e^s(b_e)\downarrow$ (and φ_e^s is currently consistent on its domain); and

(iii) $\varphi_e^s(a_e)\,|_B\,\varphi_e^s(b_e)$.

The action is to then make $a_e\ {}_D{>}b_e$ by moving the f_{s+1} partial isomorphism around, using the density of D_2 to do this without disturbing our earlier work for any other R_j. $\square$

The ideas above are elaborated and used in quite a number of the proofs of Manaster and Remmel [138], giving results such as

Theorem 7.18 (Manaster and Remmel [138]) *(i) $\mathbb{Q}^2$ is decidably categorical (i.e. any two decidable presentations of $\mathbb{Q}^2$ are computably isomorphic) and so is D_2. Furthermore if P is a decidably presented dense 2dpo then P is computably isomorphic to a computable subset of a decidable presentation of $\mathbb{Q}^2$.*

(ii) $\mathbb{Q}^2$ is not computably categorical but D_2 is (so all computable presentation of D_2 are decidable ones).

(iii) $D_2^{1,0}$ is decidably categorical but not computably categorical.

(iv) For any p and q with $p+q>0$ there exists a computably rigid computable presentation of $D_2^{p,q}$. (Note that $D_2^{p,q}$ has $2^{\aleph_0}$ classical automorphisms; and D_2 is not computably rigid (via Cantor)).

We remark that the proof of (iv) requires a much more elaborate finite injury priority argument. The positive results above are proven by back-and-forth arguments. For more on dense 2dpo's the reader is referred to Manaster and Remmel [138, 139].

Remmel [179] and Metakides and Remmel [151] have some other work associated with dense 2dpo's. Again this is rather outside the scope of this paper, so we restrict ourselves to a few brief comments with respect to the flavour of this work.

The computably enumerable sets form a lattice $\mathcal{E}$ which has been extensively studied. The main approach is to study the quotient lattice $\mathcal{E}^*$, that is, $\mathcal{E}/{=^*}$ where $A =^* B$ iff $|(A-B)\cup(B-A)| < \infty$. One seeks to understand the relationships between the structure of $\mathcal{E}$ and the computably enumerable degrees. On the other hand, when we look for the analogue of $\mathcal{E}^*$ in other computable structures, $=^*$ is often not the correct notion. For instance, in the lattice of subspaces, the analogue of $=^*$ is "= modulo a finite dimensional piece".

Metakides and Remmel [151] introduced a notion (*in a general position (i.g.p.)*) — corresponding to the notion of infinite/coinfinite in $\mathcal{E}$ which pertained to a wide class of decidable models M including (eg) $\mathbb{Q}$ and $D_2^{m,n}$ (and $\mathbb{N}$). In this paper and subsequently Remmel [179] a number of satisfying extensions of results for $\mathcal{E}$ were obtained. We give only one example. A set A in the model M is said to be i.g.p. if for every infinite set $X \subseteq \mathbb{N}$ definable from finitely many parameters of M, both $X \cap A$ and $X - A$ are infinite. A set $A \subseteq M$ is called *cohesive* (i.g.p.) if A is i.g.p. and for any computably enumerable set $W \subseteq M$ either $W \cap A$ or $A - W$ is not i.g.p. (in $\mathcal{E}$ a set A is cohesive if for all computably enumerable sets W either $|W \cap A| < \infty$ or $|A - W| < \infty$).

Theorem 7.19 (Metakides and Remmel [151]) *If M is appropriate, as above, then M has a co-computably enumerable cohesive set i.g.p. (this generalises the well-known result of Friedberg [68] for $\mathcal{E}$).*

Returning now to the main line of this section, we have seen that computable dimension and dimension are quite different. A famous classical theorem of Dilworth [44] asserts that if D is a poset with finite width n (i.e. the largest antichain has size n) then D can be written as the union of n disjoint chains (and, indeed, the dimension of D is thus $\leq n$).

Following Kierstead, McNulty and Trotter [116], we call the least n such that D can be written as the union of n disjoint computable chains the *computable chain covering number*. The exploration of the connections between these notions leads to some fascinating and rather complex combinatorics. Kierstead et al [116] improved the Manaster-Rosenstein result Theorem 7.11 to show

Theorem 7.20 (Kierstead, McNulty and Trotter [116]) *There is a computable ordered set A of width 3 with computable chain covering number 4 and no computable (finite) dimension.*

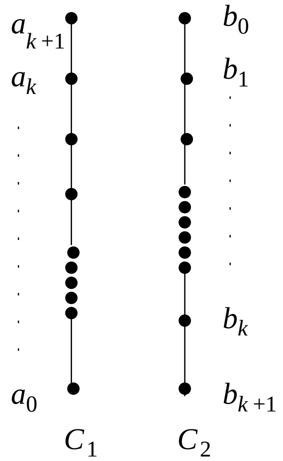
One of the key notions we will need is that of a *computable realizer*. We say that $(L_1, \ldots, L_n)$ is a (*computable*) *n-realizer* for a computable P if $P = L_1 \cap \cdots \cap L_n$ with the L_i all computable linear orderings.

Proof To prove Theorem 7.20 we must build a computable ordered set A of width 3 and computable chain cover number 4 to meet

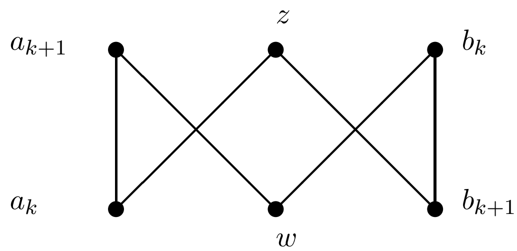
$$R_e \qquad A \neq \cap_{i \in D_e} L_i$$

where L_i denotes the ith computably enumerable linear ordering and D_e the eth canonical finite set. That is, we diagonalise over all computable realizers.

For a single R_e we proceed as follows. We begin to build chains C_1 and C_2 as described in the diagram below.



Following Kierstead, McNulty and Trotter [116] define pairs $(x,\, y)$ and $(p,\, q)$ to be *pointlike* (relative to D_e) if $x < y$ and $p < q$ (in A) and for all $i \in D_e$ either $q <_i x$ or $y <_i p$ where $<_i$ denotes the order relation on L_i. When we add a_{k+1} and b_{k+2} we wait to see if the pair $(a_k,\ a_{k+1}),\ (b_k,\ b_{k+1})$ is pointlike before we define a_{k+2} and b_{k+2}. The goal is to find a pointlike pair $(a_k,\ a_{k+1}),\ (b_k,\ b_{k+1})$. If we find such k we meet R_e as follows. We enumerate $z \in C_3$ and $w \in C_4$ as follows



Now suppose in some L_i we have $z <_i w$. then as the pairs are pointlike, either $b_k <_i a_k$ or $b_{k+1} <_i a_{k+1}$. If, for instance, $b_k <_i a_k$ we have $z <_i w <_i b_k <_i a_k <_i z$. From this it follows that in any L_i we must have $w <_i z$ and hence $A \neq \cap_{i \in D_e} L_i$.

It suffices to show for a single R_e that a pointlike pair must exist. After the pair $(a_j,\ b_j)$ is added, let m_j denote the number of L_i with $i \in D_e$ such that $a_j\ {}_i\!>\! b_j$. Then assuming we are consistent with $A = \cap_{i \in D_e} L_i$ we must have $1 \leq m_j \leq |D_e| - 1$ for all j. This follows since $a_j \,|\, b_j$ so $a_j\ {}_i\!>\! b_i$ for some i, but not all i. (Of course if this fails, we simply stop and win R_e outright.) We know that $a_{j+1} > a_j$ and $b_j > b_{j+1}$ in A. Hence if $a_j\ {}_i\!>\! b_j$, we have $a_{j+1}\ {}_i\!>\! b_{j+1}$. Thus $m_j \leq m_{j+1}$ and hence, by the pigeonhole principle, for some $k < |D_e|$ we have $m_k = m_{k+1}$. For this k, $(a_k,\ a_{k+1}),\ (b_k,\ b_{k+1})$ is pointlike.

To complete the proof, it suffices to say how to combine the R_e's. But this is easy. Split the construction into infinitely many levels with R_e working (locally) at level e (which can be thought of as a box in which we meet R_e). We declare the boxes ordered in A via box $e <$ box f if $e < f$. Finally note that $C_1 \cup C_2 \cup C_3 \cup C_4 = A$. $\square$

At this stage, we remark that the results of Kierstead et al [116] and Kierstead [110, 113] all had proofs cast in the language of computable game theory. Here we choose not to do this since we would like to give an alternative view. Most (all) of the constructions of recursion theory or computable combinatorics can be recast in this way, but it is not clear if such an approach yields greater insight. There is some evidence each way. We remark that Theorem 7.20 leads to a fundamental question.

Question 7.21 *What are the precise relationships between $W(A)$, the width of A, $C(A)$, the computable covering number of A and $d(A)$, the computable dimension of A?*

The reader should recall that in Manaster and Rosenstein [141] produced a computable 2dpo that was not even isomorphic to a computable subset of $\mathbb{Q}^2$. Theorem 7.20 leaves open the following question.

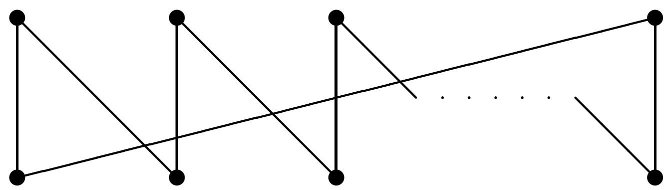
Question 7.22 *Is there a computable A of finite width such that A is not isomorphic to a computable subset of $\mathbb{Q}^n$ for any n? What about computable ordinals n?*

Also

Question 7.23 *(i) For a given n what values can $d(A)$ assume for A of computable width n? Is each $m \geq n$ possible?*

(ii) What happens if A is dense?

Kierstead et al [116] observed that the poset of Theorem 7.20 has many induced suborderings that are 3-crowns. Here an n-crown is a collection of $2n$ points forming the following Hasse diagram.



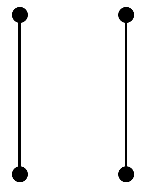
Using an elaboration of the idea of pointlike pairs of Theorem 7.20, these authors showed

Theorem 7.24 (Kierstead, McNulty and Trotter [116]) *Let $c > 1$. There exists a computable crown-free poset A with $c(A) \leq c$ such that $d(A) \geq c \binom{c-1}{t}$ where $t = \lfloor (c-1)/2 \rfloor$.*

Kierstead [113] points out that it is not clear if one needs to exclude only 3-crowns in Theorem 7.24.

Question 7.25 *Is there any condition we can put on A to ensure, for instance, $d(A) \leq w(A)$ or $d(A) \leq c(A)$ or $c(A) = w(A)$? Also if $c(A) = w(A)$ is it true that $d(A) < \infty$?*

One class of orderings that has attracted some attention are *interval orderings* since these form a relatively well understood subclass of the crown-free orderings. For the reader unfamiliar with the notion, one characterisation is that P is an interval ordering iff P has no subordering of the type given below



The name actually comes from the manner in which such orderings can be represented. In the negative direction, Hopkins improved the results of Kierstead et al [116] for interval orders:

Theorem 7.26 (Hopkins [94]) *There are computable interval orders of width w and computable dimension $\lceil \frac{4}{3}w \rceil$.*

Proof sketch Let A be an interval ordering. Define, for $x \in A$,

(i) $U(x) = \{y : x < y\}$ and $D(x) = \{y : y < x\}$, and

(ii) $U_*(x) = \{y : U(x) \subset U(y) \text{ and } x \neq y\}$ and

(iii) $D^*(x) = \{y : D(x) \subset D(y) \text{ and } y \neq x\}$.

We call an element down (up) in a linear extension L if x is less (greater) than all elements of $D^*(x)$ ($U_*(x)$) in L. Given w, we must build A of width w to meet

R_e If $t = \lceil \frac{4}{3}w \rceil - 1$ and $|D_e| = t$, then $A \neq \cap_{i \in D_e} L_i$.

For the discussion of a single R_e, we will simply let $D_e = \{1, \ldots, t\}$.

The initial play is to build $\frac{1}{2}w$ element antichains A and B with $B < A$. Now we wait until $L_1 \cap \cdots \cap L_t$ covers these elements. Define $f(m)$ ($g(m)$) to the unique element of $A(B)$ that is down (up) in L_m. Now we appeal to the following combinatorial hashing lemma.

Lemma 7.27 (Trotter and Munroe [220]) *If f and g are functions from $\{1, \ldots, t\}$ to $\{1, \ldots, w\}$ where $t = \lceil \frac{4}{3}w \rceil - 1$, then there exist $i, j \in \{1, \ldots, w\}$ such that $f(m) = i$ and $g(n) = j$ imply $m = n$.*

By Lemma 7.27 there are $a \in A$, $b \in B$ and m with a only down in L_m and b only up in L_m. Then we win by putting a new point q below all elements of A except a and above all elements of b except b. Then in L_m it must be that $c < a$ and $b < c$, this is a contradiction.

The strategies again fit together in the obvious way. □

Before we look at any further negative results, we will now examine a rich collection of very surprising positive results. We begin with Kierstead's effective version of Dilworth's theorem.

Theorem 7.28 (Kierstead [110]) *A computable ordered set A of width m has computable chain cover number $\leq (5^m - 1)/4$.*

Proof The argument is quite complex and is described in great detail in [110]. Another account is given in Kierstead [113]. We will only sketch the details following Kierstead [113]. The argument is proven by induction, and the strategy is given by recursion.

Let $n = (5^m - 1)/4$, and suppose the result for all computable orderings of width $\leq m$. Suppose A has width $m + 1$. Let $k = (5^{m+1} - 1)/4 = 1 + 5(5^m - 1)/4$. We need to construct chains $C_1, \ldots, C_k$. These we label as $B, C_{i,j} : 1 \leq i \leq n, 1 \leq j \leq 5$. At stage s, the idea is to try to put s into B. If we cannot do this, we put s into a set Q. On this set Q there is a width m computable order R extending A's orderings O which we define. This ordering R has certain special properties. We appeal to the induction hypothesis to use the strategy for width m orderings to put s into one of the n R-chains $D_1, \ldots, D_n$. This is okay by the uniformities of the proof. Finally the special properties of R will ensure that if s is put into D_j then s is put in one of $C_{j,1}, \ldots, C_{j,5}$ which cover D_j.

Property 7.29 *The exact properties we need for R are that it extends A, and (at stage s, dropping the s's).*

(i) (Q, B) is a partition of A with

- *(a) B is a maximal O-chain*
- *(b) $O \,|\, Q \subseteq R \,|\, Q$*
- *(c) $D_1, \ldots, D_m$ covers Q*
- *(d) if $x <_R z$ and $x \,|\, b \,|\, z$ in O, for some $b \in B$ enumerated before the latter of x and z, we have $x < z$ in O. Furthermore, if $x < y < z$ in R then $y \,|\, b$ in O.*

(ii) $(Q, \sim)$ is an equivalence relation such that

- *(a) $x \sim y$ implies ($x < y$ or $y < x$ in O).*
- *(b) The equivalence classes of $(A, \sim) \,|\, D_j$ are convex in R for all j.*

(That is, $x, y, z \in D_j$, $x \sim z$ and $x < y < z$ implies $x \sim y$ and $y \sim z$.

(iii) For each j, $C_{j,1}, \ldots, C_{j,5}$ is a chain cover of D_j in O such that if $x \sim y$ then $x \in C_{j,k}$ iff $y \in C_{j,k}$.

To ensure that these properties hold at each stage the induction goes through, the play is as follows.

Step 1 Given $Q_s, B_s, R_s, D_1^s, \ldots D_m^s$ i.e. $Q, B, R, D_1, \ldots, D_m$ at stage s, we need to place s. If $B_s \cup \{s\}$ is a chain, put s into B_{s+1}. Otherwise put s into Q_{s+1}. Now let $x < s$ ($s < x$) in R_{s+1} iff one (or more) of the following holds:

(i) $x < s$ ($s < x$) in (A, O)

(ii) For all $b, c \in B$ if $b \,|\, x$ and $c \,|\, s$ in O then $b < c$ ($c < b$) in O.

(iii) $(\exists a \in Q)(x < a$ ($a < x$) in R_s and (i) and (ii) above hold with x replaced by a).

Now use the inductive strategy to put s into one D_j. (Note the verification requires us to show that the width of Q_{s+1} remains at m).

Step 2 Next we need to construct $\sim_{s+1}$, $C_{i,j}^{s+1}$ for all $1 \leq i \leq n$, $1 \leq j \leq 5$. Assume $s \in D_j^{s+1}$. Let s^- (s^+) be the largest (smallest) element D_j^{s+1} less than (greater than) s in R_{s+1}. Extend $\sim_s$ to $\sim_{s+1}$ via

(i) s is in the equivalence class of s^- if there exists $b \in B_{s+1}$ with $s \,|\, b \,|\, s^-$ in $(A,\, 0)$.

(ii) s is in the equivalence class of s^+ if (i) does not pertain *and* there is $b \in B_{s+1}$ with $s \,|\, b \,|\, s^+$ in $(A,\, 0)$.

(iii) If neither (i) nor (ii) pertain, then s is in a new equivalence class.

If $s \sim_{s+1} x$ and $x \in C_{j,k}^s$ put s into $C_{j,k}^{s+1}$. Otherwise put s into $C_{j,e}^{s+1}$ where e is chosen so that neither of the two equivalence classes over s in $(Q_{s+1},\ R_{s+1}) \restriction D_j^{s+1}$ nor those immediately below s in $(Q_{s+1},\ R_{s+1}) \restriction D_j^{s+1}$ are contained in $C_{j,e}$.

To complete the proof one must give an exact analysis of the construction to show that the properties 7.29 are indeed preserved by the construction. The details are not difficult, but don't really give any further insight save to note that the conditions 7.29 specifically chosen to allow the induction on stage number to proceed. We refer the reader to, in particular, Kierstead [113, page 87] where these are all spelled out. $\square$

Szemeredi gave a lower bound for Theorem 7.28 via the following which examines the effective content of Mirsky's [153] dual to Dilworth's theorem.

Theorem 7.30 *(i)* **(Schmerl (see Kierstead [113])** *Every computable ordering of finite height h (ie the longest chains is size h) can be covered by $\binom{h+1}{2}$ computable antichains.*

(ii) **(Schmerl (see Kierstead [113])** *For all h there is a computable ordering of height h that cannot be covered by fewer than $\binom{h+1}{2}$ computable chains.*

(iii) **(Szemeredi (see Kierstead [113])** *In (ii) we can choose the counterexamples to have computable dimension 2.*

Proof

(i) Let P have height h and let $n = \binom{h+1}{2}$. We construct antichains $A_1, \ldots, A_n$ to cover P. Label the antichains as $A_{d,u}\ :\ 0 \leq d+u \leq h-1$. At stage s, put s into $A_{d,u}^{s+1}$ where d is the length of the longest chain in P_{s+1} strictly below s and u is the length of the longest chain strictly above s. To complete the proof we need only argue that $A_{d,u}^{s+1}$ is an antichain. Suppose, for instance, $s < x$. Then there is a chain of $u+1$ elements strictly above x. So $s \notin A_{d,u}^{s+1}$ after all.

(iii) (sketch) We must build $(P,\, R)$ with computable realizer $(L_1,\, L_2)$ to diagonalise over all sets $A_1, \ldots, A_n$ of computable antichains with $n = \binom{h+1}{2} - 1$, and this is done by induction on h. Let $h = g+1$. For a particular $A_1, \ldots, A_n$ we proceed as follows, again following Kierstead's account [113]. We concentrate on a single requirement.

At stage s, the construction is in 2 steps, with $P = P_1 \cap P_2$ where P_i are the sets of points added in step 1. In step 1 our goal is to arrange that the top h points T_s of P_1^s in L_1^s are all incomparable in R_s yet we force all the elements of T_s into different antichains.

To achieve this, suppose the points Q_r^s from P_1^s have been played so that $P \mid Q_r^s$ has height $\leq r$, that the top r points T_r^s of Q_r^s in L_1 are all incomparable in R and we've forced exactly r antichains from $A_1^s \cdots A_n^s$ to cover Q_r^s, each of which has an element of T_r^s. Now we add new points $c_1, \ldots, .c_j$ into P_1^s one at a time so that c_i is the $r+1$'st point of L_1^s and the top point of L_2^s (currently) until the requirement is forced to put some c_j into A_k^s. This occurs by the point when c_j is in a chain of length $r+1$ as $c_1 \cdots c_j$ is a chain. In this way via the substeps r we get to the first goal.

In the second step we ensure that $(P_2^s, R \restriction P_2^s)$ has height g, every element of P_2^s is under every element of T_s in R_s and incomparable in R_s to all elements of $P_1^s - T_s$; finally at least $\binom{g+1}{2}$ computable antichains are needed to cover P_2^s. It then follows that we can get $A_1 \cdots A_n$ since we'll need $h + \binom{g+1}{2}$ of them to cover and $h + \binom{g+1}{2} = \binom{h+1}{2}$.

To achieve this second goal, we simply use the inductive strategy we used for g, defining the new ordering in the obvious way: all new points x in P_2^s are over ones in $P_1^s - T^s$ and under all points in T_s in L_1^s and putting x under all points of P_1^s in L_2^s.

Again, there is no problem with using the layering technique here so the strategies combine as usual. □

Corollary 7.31 (Szemeredi (1982) (in Kierstead [113])) *For any m there is a computable ordered set of width m that cannot be covered by fewer than $\binom{m+1}{2}$ computable chains.*

Proof We use Theorem 7.30(iii). Let (P, R) be computable dimension 2 computable ordered set with computable antichain number $\binom{m+1}{2}$. Define P^* to be $(P, L_1 \cap L_2^*)$ where L_2^* is the dual of L_2. A is a chain (antichain) in P^* iff A is an antichain (chain) in P. The width of P^* is m and P^* cannot be covered by $< \binom{m+1}{2}$ computable chains. □

Using a different argument, Kierstead improved Corollary 7.31 for $m = 2$.

Theorem 7.32 (Kierstead [110, 113]) *There is a computable poset of width 2 which cannot be covered by fewer than 5 computable chains.*

Other results in this area complement our earlier negative ones as follows:

Theorem 7.33 (Kierstead, McNulty and Trotter [116]) *Every computable crown-free ordering which can be covered by at most c computable chains has computable dimension at most $c!$ (compare with Theorem 7.24).*

Theorem 7.34 (Kierstead and Trotter [115]) *Every computable interval order of width m can be covered by $3m-2$ computable chains, and there are computable interval orders that cannot be covered by fewer than $3m-2$ computable chains.*

Theorem 7.35 (Hopkins [94] (see Kierstead [113])) *Every computable interval order of width m has computable dimension at most $4m-4$ (compare with Theorem 7.26).*

The arguments above, particularly Theorems 7.33 and 7.34 are quite involved and we don't have space here to give a fair treatment of them. The reader should consult particularly Kierstead [113] for details. One last result we will prove is the following that says the width, computable chain cover number c and dimension are related if the computable dimension is bounded.

Theorem 7.36 (Kierstead, McNulty and Trotter [116]) *If (P, R) is a computable ordered set with $d(P) = d$ and P has width m, then $c(P) \leq \binom{m+1}{2}^{d-1}$.*

Proof Use induction on d. Let $d = e + 1$ and $L_1, \ldots, L_{e-1}$ a computable realizer of P. Let $Q = L_1 \cap \cdots \cap L_e \cap L^*_{e+1}$ with L^*_{e+1} the dual of L_{e+1}. Then Q has height bounded by m. By Schmerl's result, A can be covered by $\binom{m+1}{2}$ computable antichains so P can be (in Q). If A is a computable antichain in Q then $P \upharpoonright A$ is a computable ordered set of width $\leq m$ and has realizer $(L_1 \upharpoonright A, \ldots, L_e \upharpoonright A)$. Now we can use the induction hypothesis to cover each of the $\binom{m+2}{2}$ computable antichains covering P by $\binom{m+1}{2}^{e-1}$ computable chains. □

We close with a few comments and a list of further questions. First Kierstead pointed out that there is a huge gap between Szemeredi's quadratic lower bounds and the exponential upper ones (for $c(P)$).

Question 7.37 *(i)* **(Kierstead)** *Let $b(m)$ be the maximum $c(P)$ over all computable P of width m. Is $b(m)$ polynomial? What is $b(2)$?*

(ii) **(Kierstead, McNulty and Trotter)** *Is $d(P)$ finite if P has finite width and is 3-crownfree?*

Another possible approach is to examine dimension and covering numbers in terms of the arithmetical hierarchy. For instance, we would define the Π_n dimension (of A) to be the least m with $A = L_1 \cap \cdots \cap L_m$ and the L_i all Π_n linear orderings. (Similarly we could do this for covering number.)

Question 7.38 *Investigate this notion. For example, is there a finite n with the Π_n dimension of an ordering the same as the classical dimension?*

Question 7.39 *How do the (degrees of) realizers relate to known notions in recursion theory such as Π^0_1 classes, etc. (eg Jockusch-Soare [98]) and to reverse mathematics? (See Simpson [204]; for related results see Hirst [91].)*

8 Degree-Theoretical Results, Jump Degrees and Worker Type Arguments

We return now to the problem of assigning a degree to the isomorphism type of a linear ordering. The natural approach is to define the degree of an order type to be the minimum of the degrees of the orderings of that type. By Richter's [188] result we met in Section 3 we know that if an ordering has a degree $\mathbf{a}$ then $\mathbf{a} = \mathbf{0}$. There are a number of possible extensions to this idea of degree which are useful for order types, and we shall look at some in this section. The first was suggested by Jockusch.

Definition 8.1 Let $\mathbf{A}$ be a structure. Define the *n-(jump) degree* (n a computable ordinal) of $\mathbf{A}$ to be $\min\{(\deg(\mathbf{B}))^{(n)} : \mathbf{B} \cong \mathbf{A}\}$ where $\mathbf{a}^{(n)}$ denotes the n-th jump of $\mathbf{a}$.

We shall say an ordering A has *proper n(-th jump) degree* $\mathbf{a}$ if A does not have k-degree for any $k < n$ and A has n-degree $\mathbf{a}$. Julia Knight asked if a linear ordering could have proper 1-degree $\mathbf{0}'$. The reader should note that this is of some technical interest since the Feiner coding techniques of Section 3 (see remark 3.8) construct Π_1 orderings not isomorphic to *low* orderings. Here the reader should recall that a set A (degree $\mathbf{a}$) is *low* if $A' \equiv_T \emptyset'$ (resp. $\mathbf{a}' = \mathbf{0}'$). This question was answered affirmatively by Jockusch and Soare, who introduced a new idea into these studies: the idea of separators with requirements played between them. This idea has some reflections in some of the earlier work of Watnick and Lerman but is rather different.

Theorem 8.2 (Jockusch and Soare [101]) *(i) If B is computably enumerable and $B \not\equiv_T \emptyset$, then there exists a linear ordering A with $A \equiv_T B$ such that if $C \cong A$ then $C \not\equiv_T \emptyset$.*

(ii) In particular, if B is chosen low then it follows that there is an A of proper 1-degree $\mathbf{0}'$.

Proof We first sketch the proof without permitting (which keeps $A \leq_T B$). We build a Π_1 ordering A ($\subseteq \mathbb{Q}$) so that

$$R_e : A \not\cong W_e.$$

First we concentrate on the basic module (for a single R_e). To meet R_e, we play a game with W_e. We first enumerate a point a_0 in A and wait till there occurs some point b_0 in W_e. Now we proceed as follows. We begin to build an ω^* sequence in A_e, $a^s_{n(s)} < \cdots < a^s_2 <^s a^s_1 <^s a_0$ until W_e responds with a point $b_1 < b_0$. Note that if no such b_1 occurs in W_e then we win R_e as every point in A has infinitely many predecessors yet b_0 does not in W_e. Now if W_e responds at stage s_1 with $b_1 < b_0$ in W_{e,s_1}, we define *all of $a^s_{n(s)}, \ldots, s^s_1, a_0$ to be equal and define $a^{s_1}_1 > a_0$ and begin a new ω^* sequence in A immediately above a_0 but below a^t_1* as below in Figure 9

$$a_0 <^t a^t_{n(t)} <^t \cdots <^t a^t_1.$$

Figure 9:

Again we wait for there to occur some $b_2 < b_0$ in W_{e,s_2} for some $s_2 \geq s_1$. If no such b_2 occurs we win as A has order type $1 + \omega^*$ yet b_0 is not the last point of W_e but it has only one predecessor.

The inductive strategies continue in the obvious way. That is, at any stage q we have

$$a_0 < a_1 < a_2 < \cdots < a_{k-1} < a^q_{n(q)} < \cdots < a^q_{k+1} <^q a_k.$$

Now if ever W_e does not respond then we build a $(k-1) + \omega^*$ sequence for A and win as b_0 has k predecessors. If W_e *always* responds then we win as *we* build an ω sequence *yet* b_0 *has infinitely many predecessors.*

The above concludes the basic module. Now to put the actions together for many R_e, Jockusch and Soare came up with the novel idea of a static set of *separators*. The final order type of A will be of the form

$$A_1 + S_1 + A_2 + S_2 + \cdots$$

where S_i is $1 + \eta + 1 + i + 1 + \eta + 1$. Now if $W_e \cong A$ then we must have S_i for all i. The idea is that we use the interval between S_{e-1} and S_e to meet R_e. Suppose $W_e \cong A$. Let $(a,\, b,\, c,\, d)$ represent the separator S_i in W_e. That is, $(a,\, b,\, c,\, d)$ are the points with a the left 1, $(b,\, c)$ the $i + 2$-block and d the right 1. Then the relation

$$R(a,\, b,\, c,\, d) = \text{“}(a,\, b,\, c,\, d) \text{ represent } S_i \text{ in } W_e\text{”}$$

is a Π_2 relation uniformly in i, and so we can handle it on a Π_2 tree of strategies, as in Watnick's theorem of Section 7.

Thus for the e-module we attempt to do the basic module between S_{e-1} and S_e. Now in W_e we guess the quadruples q_1 and q_2 (or more neatly, the pair $(q_1,\, q_2)$) corresponding to S_{e-1} and S_e. Based on this belief we proceed as above, building A. The secret is that if ever a guess appears wrong (via a higher priority guess appearing correct) we can always cancel our mistakes: after all A is not computable so we can declare all the points based on weaker guesses to be equal. As usual, guesses that are too strong give only finitely many points wrong and by this observation the others don't matter so the tree $q_1,\ q_2$ can be used to win, if they exist.

We keep $A \leq_T B$ as follows. We only ask that if $B_s[k] = B[k]$ and $j < k$ then $a_j <^s a_k$ implies $a_j < a_k$ in A. That is, we can only change the ordering if the indices are permitted. This is an essentially routine modification. For instance, for a_0 we build our ω^* sequence at each stage s,

$$a_{n(s)} <^s \cdots <^s a_1 <^s a_0$$

until we see some $m < n(s)$ with $b_m < b_{m-1} < \cdots < b_e < b_0$ and $B_{s+1}[m] \neq B_s[m]$. Then we cut the a_0-predecessors back to

$$a_{m-1} <^{s+1} \cdots <^{s+1} a_0 <^{s+1} \cdots <^{s+1} a_m$$

. This clearly causes no real hardships and we need only note that if points are placed for α and later for β and $\alpha <_L \beta$ then the points for β are bigger than those for α. Thus there is no real interference caused by the various e-modules between S_{e-1} and S_e. One point that should be noted is that the R_e-module must act at other than stages when it looks correct, in the "cutback" sense. We must only ensure that new points are defined only at stages when it looks correct.

The proof outlined above only gives $A \leq_T B$. To get $A \equiv_T B$, Jockusch and Soare proved the following:

Theorem 8.3 (Special case of Knight [118, Theorem 4.1]) *Let B be any set and suppose A is a linear ordering with $A \leq_T B$. Then there is a $C \cong A$ with $C \equiv_T B$.*

Proof (Jockusch and Soare [101]) We code B into C. To do this we ensure that, for all k, $c_{2k} <_C c_{2k+1}$ iff $k \in B$. Define an isomorphism f via: if ($k \in B$ iff $a_{2k} <_A a_{2k+1}$) let $f(a_{2k}) = c_{2k}$, $f(a_{2k+1}) = c_{2k+1}$; otherwise, let $f(a_{2k}) = c_{2k+1}$ and $f(a_{2k+1}) = c_{2k}$. □

The reader might wonder if the hypothesis that B has computably enumerable degree is necessary in Theorem 8.2. If a set $B \leq_T \emptyset'$, we know that it has a computable approximation by the limit lemma. This leads to the technique of "Δ_2 permitting". We have already seen this in the proof that if A is a Δ_2 linear ordering then A is isomorphic to a Π_1 one. However, we recall what this means. We have $B = \lim_s B_s$ with $B_s(x) \neq B_{s+1}(x)$ only finitely often. We then construct $C \leq_T B$ by simple Δ_2 permitting by asking that if $B_s[x] = B[x]$ then $C_s[x] = C[x]$. Basically this means that if we think we have a permission we can do something. Then if later this permission is withdrawn, then we must "go back". We must ensure that the construction works in the "true stages" of b's enumeration. The details are a little messy but, using this idea, the author, and, independently, D. Seetapun showed the following:

Theorem 8.4 (Downey, Seetapun) *If* $\mathbf{a}$ *is a* Δ_2 *nonzero degree then* $\mathbf{a}$ *contains a linear ordering isomorphic to a computable one.*

Julia Knight improved some earlier work of Ash, Knight and Jockusch to show that a consequence of Theorem 8.4 is

Theorem 8.5 (Knight) *If* $\mathbf{a} \neq \mathbf{0}$ *is any degree then* $\mathbf{a}$ *contains a linear ordering not isomorphic to a computable one.*

Proof Let C be given. We show how to construct an ordering computable from C not isomorphic to a computable one. The proof splits into 3 cases.

Case 1. $\emptyset'' <_T C''$. This case uses some work of Ash, Jockusch and Knight [8]. (or of Lerman [130]) Define the $\sigma(X)$ to be the "shuffle sum" of X consisting of densely many copies of $\mathbb{N}$ and of n for $n \in X$. Then it turns out that $\sigma(C'' \oplus \overline{C''})$ can be chosen to have degree X (using a proof similar to Watnick's Theorem), and cannot be isomorphic to a computable linear ordering since $\emptyset'' <_T C''$. (See Ash, Jockusch, Knight [8] for details.)

Case 2. $C \leq_T \emptyset'$ *and is noncomputable.* This is the Downey-Seetapun Theorem.

Case 3. $C'' \equiv_T \emptyset''$ *and* $\emptyset' <_T C'$. Using Case 2 in relativized form, we see that C' contains a linear ordering A not isomorphic to a computable one. Then $\gamma(A)$ can be chosen to have degree C and cannot be isomorphic to a computable ordering. □

The reader should note that the Jockusch-Soare Theorem produces a Π_1 linear ordering. Richter's result also shows that if A is any Δ_2 linear ordering with $A \not\equiv_T \emptyset$ then there is a Δ_2 linear ordering B with $B \cong A$ and $B \;_T\!\not\geq A$. The only hope therefore of any degree for Π_1 linear orderings is perhaps to define the Π_1 *degree of* A to be $\min\{\deg(B) : B \text{ is a } \Pi_1 \text{ linear ordering and } B \cong A\}$.

Theorem 8.6 (Downey (see Downey and Moses [53])) *If* $\mathbf{a}$ *is any computably enumerable degree then there is a* Π_1 *linear ordering with* Π_1 *degree* $\mathbf{a}$.

Proof This was not actually written out in Downey and Moses [53], but we give a sketch here. This uses a variation on the technique of Theorem 8.2. We build $A = \cap_s A_s$ with

$R_e : \overline{W}_e \cong A$ implies $B \leq_T \overline{W}_e$

where B is a given computably enumerable set. To make life easier, we replace B by $C = \cup_s C_s$ defined as follows. Let $f(\omega) = B$ be a 1-1 computable function. At each stage s, let $c_{i,\,s}$ list $\overline{C}_s$ in order. Define $C_{s+1} = C_s \cup \{c_{f(s),\,s},\, \ldots,\, c_{f(s)+s,\,s}\}$. Then $C \equiv_T B$ and has nice properties ("retraceability" is the one we are interested in).

For a single R_e, we build a reduction $\Gamma(W_e) = A$ if we fail to ensure that $\overline{W}_e \not\cong A$. We make $A \equiv_T B$ so this does the job. Again we begin by building an ω^* sequence below a_0. Until $\overline{W}_{e,\,s}$ "responds appropriately" we then act in the *coding mode*. That is, we have $a_{n(s),\,s},\ \ldots,\ a_{j,\,s},\ a_{j-1,\,s},\ \ldots,\ a_{0,\,s}$ and have $a_{j,\,s}$ the coding location of j. Now if j occurs in $C_s - C_{s-1}$ (and so all y in C_s with $s \geq y \geq j$), then we set $a_{j,\,s} = a_{j-1,\,s}$. (We assume $0 \notin C$.) Clearly if $\overline{W}_{e,\,s}$ does not act and this is the only action, then this section is $\equiv_T B$. Since some W_e are empty, this keeps $A \geq_T B$ via Γ as we will see.

Now as in Theorem 8.2 we need do nothing different until some j and k occur with $j < k$ and

$$b_k^s < \cdots < b_0^s \in \overline{W}_{e,\,s} \quad \text{and} \quad j \in C_s - C_{s-1} \quad \text{with } j \text{ least.}$$

If $(j,\,k)$ occur, then as in Theorem 8.2 we begin our new ω^* sequence to the right of a_0. Obviously if this repeats without interruption, then we must win since $\overline{W}_e \not\cong A$. The trouble is that in our case, $\overline{W}_e$ is not necessarily computable *and it too can change.*

First we ask that whilst it appears we've achieved a $(j,\,k)$-action as above, say at stage s, and at stage $t > s$ we see some $i < j$ occur in $C_t - C_{t-1}$ then even if $b_k^t = b_k^s$, we will still cut the sequence down to $a_{i-1}^t \cdots a_0$ and begin a_i^t to the right of a_0 as this action is more likely to succeed. (Note that for $q \geq j$ we are defining $a_q^{t-1} = a_0$ in this case.) See Figure 10.

$$a_{j-1}^{t-1} \cdots a_0\ a_{n(t)}^{t-1} \cdots a_j^{t-1}$$

becomes

$$a_{i-1}^{t} \cdots a_0\ a_{n(t)+1}^{t} \cdots a_i^{t}$$

Figure 10:

We can only be interrupted as follows. It may be that b_d^s enters $W_e - W_e^s$ for (many) $d \leq k$. A typical scenario is given in Figure 11

$$\underbrace{b_2^u < b_1^u < b_o}$$

back to 2 below b_0 in $\overline{W}_{e,\,u}$

$$\underbrace{a_{4,\,u}\ a_{3,\,u}\ a_{2,\,u}\ a_{1,\,u}\ a_0}_{\text{unchanged, perhaps since stage } s} \quad a_{150,\,u}\ \cdots\ a_{5,\,u}$$

Figure 11:

The idea is that this gives progress on $C \leq_T W_e$. To implement this, we now go *back* to the original strategy holding $a_{150+m,\,u+1} < a_{4,\,u}$ for $m = 1,\,2,\,\ldots$ and enumerate an axiom for the reduction Γ, when $\overline{W}_e$ responds. That is, we define $\Gamma(b_{150+m}^{t_1}) = a_{m,\,t_1}$ for all $2 < m < 150$

and $\Gamma(b_j^{t_1}\ (= b_j^u)) = a_{j,\,t_1}$ for $j \leq 2$ at the least stage t_1 when $\overline{W}_{e,\,t_1}$ has ≥ 150 elements below b_0. Of course if C permits below 2 before this, we don't need to worry, and if C permits below 150 before this we can put more of the a_i's below a_4. The point is that if there are infinitely many unsuccessful attacks trying to win below a_0 in this way, then Γ will eventually witness that $W_e\ {}_T\!\geq C$. The necessary details fit together as before. We should remark that this is a different procedure than that used in Downey-Moses [53]. □

We remark that not all Π_1 linear orderings have Π_1 degrees.

Theorem 8.7 (Downey) *Given computably enumerable $C \not\equiv_T \emptyset$, there is a Π_1 linear ordering $A \leq_T C$ such that A is not computable and A has no Π_1 degree. In fact, the Jockusch-Soare ordering of Theorem 8.2 has this property.*

We build $C_e = \lim_s C_{e,\,s}$ to meet

R_e $\quad W_e \equiv_T \emptyset$ or there is a Π_1 C_e with
$C_e \cong A$ and $W_e \not\leq_T C_e$.

We split R_e into infinitely many $R_{e,\,i}$:

R_e $\quad W_e \equiv_T \emptyset$ or there is a Π_1 C_e with $C_e \cong A$ and $(\forall i)(R_{e,\,i})$ with
$R_{e,\,i}$ $\quad \Phi_i(C_e) \neq W_e$.

Let $1(e,\, i,\, s) = \max\{x\ :\ (\forall y < x)(\Phi_{i,\,s}(C_{e,\,s}\,;\ y) = W_{e,\,s}(y)\}$. We shall use Sacks preservation on $u(e,\, i,\, s) = \max\{e(\Phi_{i,\,s}(C_{e,\,s}\,;\ x)\ :\ x \leq 1(e,\, i,\, s)\}$. To do this we will need a Π_2 isomorphism to witness $C_e \cong A$. That is, we build a computable finitely branching tree of possible isomorphisms and the relevant isomorphism will be given by the leftmost path so that its graph is Π_2.

Remember the Theorem 8.2 strategy is to build A_i's between separators S_j of the form $1 + \eta + j + \eta + 1$. The idea is that we will allow $\langle e,\, i\rangle$ to move the isomorphism on some numbers x in the field of $C_{e,\,s}$ (i.e. $\mathrm{dom} A_s$) with certain provisions. So $f_s\ :\ C_{e,\,s} \to A_s$. Now the idea is that if we have $f_s(z) \downarrow$ for some $z < u(e,\, i,\, s)$ we wish to keep $z \in C_e$. The trouble of course is that $f_s(z)$ may leave A_s, i.e. $f_s(z) \notin A_{s+1}$. We have only 2 options. Either we delete z from $C_{e,\,s+1}$ or we shift the isomorphism. The obvious place to shift such a $f_s(z)$ is into the η-part of some separator. Then we will never need to delete z from C_e. Unfortunately, if we do this infinitely often, for some fixed A_i, then C_e will not be isomorphic to A since the "1" from $1+\eta+i+\eta+1$ may be missing.

To overcome this problem, we will only allow $R_{e,\,i}$ to shift f_e on A_j for $j > \langle e,\, i\rangle$. So for A_j with $j \leq \langle e,\, i\rangle$ we will take the "delete z" option. This creates a few problems. Now we only need to build $C_e \cong A$ if $W_e \not\equiv_T \emptyset$. So the argument is that such $R_{e,\,i}$ only have finite effect if $W_e \equiv_T \emptyset$. First we don't have a problem arguing that $\Phi_i(C_e) \neq W_e$ with this: Since the only A_j that use the "delete z" option (and hence potentially change C_e on $u(e,\, i,\, s)$) are those with $j < \langle e,\, i\rangle$, then the Π_2 correct version of $R_{e,\,i}$ can know exactly which of the A_j for $j < \langle e,\, i\rangle$ act infinitely often. Since we know exactly how it will act, we can wait till $F_i(C_e;\ y) = W_{e,\,s}(y)$ via j-correct computations for $j < \langle e,\, i\rangle$. Since j-correct computation can be preserved (by shifting), this means that if $W_e \not\equiv \emptyset$, then $\Phi_i(C_e) \neq W_e$.

However, the problems are created by the "false" $R_{e,\,i}$ (those with incorrect guesses). We focus on $k \leq \langle e,\, i\rangle < j$. The problem comes in trying to keep $C_e \cong A$. For instance, might have some Φ_i computation with some $z < u(e,\, i,\, s)$ for which we shift the isomorphism so that z is part of a S_j separator, We would add some new elements to pick up the A_j^s-computations (or A_{j-1} depending on which way we moved $f_s(z)$). The trouble is that some $z_1 < z$ in A_k^s, some $k < \langle e, i\rangle$, may be false. So later we remove z_1 from A_k^t. If this recurs infinitely often for the *same* Φ_i-computation (which we meet by divergence) we might fail to get an isomorphism on A_j. The idea is that when this occurs, *we delete the new points we added to* $C_{e,s}$ *and put* $f_s(z)$ *back to where it originally was.*

Using the true path method we can see that this designs a Π_2 isomorphism from C_e to A provided that we meet the $R_{e,i}$ on the true path. Suppose then that for some $R_{e,i}$ we have $\langle e, i\rangle$-correct computations. This in the view that we build a tree of potential isomorphisms and if β is the true path then f_β is the correct f. Now if we move $f_s(z)$ for an $\langle e, i\rangle$-correct computation, we won't need to move it back. If then the true version of $R_{e,i}$ acts infinitely often then we see, in the usual way, that W_e is computable (to compute $W_e(z)$ go to a stage where we have $1(e, i, s) > z$ via $\langle e, i\rangle$-correct computations). □

We remark that the above uses Π_2 isomorphisms which are different from all of our other proofs which essentially construct Δ_2 ones. It is not clear if we can modify the above proof to show

Question 8.8 *Does the Jockusch-Soare ordering of Theorem 8.2 (for any C) have the additional property that if B is a Δ_2 and isomorphic to A then there is a $\Pi_1 D \cong A$ with $B \not\leq_T D$.*

Another open question here is:

Question 8.9 *Is there a Π_1 linear ordering A with no computable copy but A has copies of each computably enumerable (Δ_2?) non-zero degree? (This seems unlikely.)*

The original use of the separator technique by Downey and Moses was to look at the degrees of successivities in a computable linear ordering A. Those authors were motivated by the following questions of Remmel:

Question 8.10 (Remmel) *If A is a computable boolean algebra, does there exists $B \cong A$ with B computable and the atoms of B of incomplete Turing degree?*

Downey [49] has shown the answer to Question 8.10 to be "yes".

The corresponding question for linear orderings asks if the successivity relation can ever be "intrinsically complete".

Theorem 8.11 (Downey and Moses [53]) *If A is computably enumerable and noncomputable, then there exists a computable linear ordering B with $S(B) \equiv_T A$, where $S(B)$ denotes the successivities of B, such that if D is a computable linear ordering with $D \cong A$ then $S(D)\ {}_T\!\geq A$. In particular if $A \equiv_T \emptyset'$ then $S(D) \equiv_T \emptyset'$.*

Proof The original proof made direct use of separators. However, armed with the Downey-Knight result Theorem 3.16, we can give a proof using Theorem 8.6. Recall that this concerned $\gamma(Q)$ where for a linear ordering Q,

$$\gamma(Q) = Q(\eta + 2 + \eta).$$

The result was that *Q is Π_{n+1} iff $\gamma(Q)$ is Π_n*. Apply this result via Theorem 8.6 to any linear ordering C of Π_1 degree $\deg(A)$. Then $B = \gamma(C)$ is computable and, by construction, $S(B) \equiv_T A$ and if $D \cong B$ is computable then $S(D)\ _T\!\geq A$ by Theorem 8.6 and the iff in Theorem 3.16. $\square$

If we similarly apply Theorem 3.16 to Theorem 8.7 we see

Theorem 8.12 (Downey) *There is a recusive linear ordering A such that $S(A)$ does not have a degree. That is $\min\{\deg(S(B))$: B is a computable linear ordering with $B \cong A\}$ does not exist. Furthermore, A can be chosen with $S(A)$ of any non-zero computably enumerable degree.*

Question 8.13 *Develop a theory of "intrinsic degree" for other relations such as the block relation $B(A)$ on computable linear orderings. Develop a model theoretic setting as in Ash-Nerode [11].*

There have been some investigations into the interaction of $\deg(B(A))$ and $\deg(A)$. For instance: using rather more elaborate techniques, based on the separator idea, Jockusch and Soare showed:

Theorem 8.14 (Jockusch and Soare [101]) *There is a Π_1 linear ordering A such that $\langle A,\ B(A)\rangle$ is of low degree, yet A is not isomorphic to a computable linear ordering.*

Jockusch and Soare claimed in [101] that there is no common generalisation of Theorems 8.2 and 8.14. That is, *there is a computably enumerable degree $\mathbf{a} \neq \mathbf{0}$ such that if A is a linear ordering with $\deg\langle A,\ B(A)\rangle \leq \mathbf{a}$ then A is isomorphic to a computable linear ordering.* No proof of this result has yet appeared in print.

Question 8.15 *Is there a property P of classical order types that guarantees that if A has type τ, then A is low and $P(\tau)$ holds, then A is isomorphic to a computable linear ordering?*

One property is known:

Theorem 8.16 (Downey and Moses [53]) *Every low (indeed semilow) discrete linear ordering A has a computable copy.*

Proof Choose the canonical choice set B for A. Namely, put z_0 into B if z_0 is the least element in A. Then put z_1 into B if z_1 is the least with z_1 in A but not in the same block as z_0, etc. If A is semilow (i.e. $\{e : W_e \cap A \neq \emptyset\} \leq_T \emptyset'$) we then define for each triple $(x,\, b,\, y)$ an computably enumerable set $W_{g(x,\,b,\,y)}$ via

$$b \in B \leftrightarrow (\forall x < b)(x \notin A \text{ and } b \in A \text{ or } (\forall y)(W_{g(x,\,b,\,y)} \cap A \neq \emptyset))$$

where g is given by the s-m-n theorem. As A is semilow we know there is a computable f such that $W_{g(x,\,b,\,y)} \cap A \neq \emptyset$ iff $|W_{f(x,\,b,\,y)}| = \infty$. Thus $x \in B$ iff $(\forall x < b)(x \notin A$ and $b \in A$ or $(\forall y)(|W_{f(x,\,b,\,y)}| = \infty))$ so that B is Π_2. By Watnick's theorem, there is a computable Q with $C_F(Q) = B$. But then $Q \cong A$. $\square$

Similarly we can ask for a property P of computable linear orderings to guarantee that if B is a computable linear ordering with property P then there is a computable $A \cong B$ with $S(A)$ incomplete.

Theorem 8.17 (Downey and Moses [53]) *If B is a computable discrete linear ordering, then there is a computable $A \cong B$ with $S(A)$ incomplete. Indeed, given any Δ_2 $C \not\equiv \emptyset$ we can arrange for $S(C)$ ${}_T\not\geq C$.*

Proof sketch Again this uses a modification of the Watnick construction. That is, we know $B \cong \zeta\tau$ for some Π_2 τ. We then construct $A \cong \zeta\tau$, with A computable, and ensure that we make $\Phi_e(S(A) \neq C$ for any e. This uses a Π_2 version of the Sacks strategy of preserving agreements. It is rather similar to arguments we have seen earlier, so we omit the details. □

We remark that by Watnick's and Feiner's results there are computable discrete A with $S(A)$ "intrinsically non-low". This last result is tight since using any easy finite injury argument one can show:

Theorem 8.18 (Downey and Moses [53]) *If A is a computable discrete linear ordering, then there exists a computable B with $A \cong B$ and $S(B)$ semilow.*

It is not clear if there is always a B with $A \cong B$ and $S(B)$ low. Finally one other direction here is to look at *strong reducibilities.* For instance:

Theorem 8.19 (Downey and Moses [53]) *If A is a computable linear ordering then there is a computable copy B of A with $S(B)$ finite or hyperimmune, and hence, by Downey and Jockusch [48], not wtt-cuppable (so certainly not wtt-complete).*

The proof of Theorem 8.17 is again an easy finite injury argument that we omit.

Returning to our original line of investigation we can ask: what are the possible n-degrees of linear orderings? Richter proved that if $n = 0$ then $\mathbf{0}$ is the only possibility. Using an interesting forcing argument with a number of applications, one of which is a similar result, Knight proved:

Theorem 8.20 (Knight [118]) *If A has 1-degree $\mathbf{a}$ then $\mathbf{a} = \mathbf{0}'$.*

The proof of Theorem 8.20 uses a forcing argument which would take us a little far afield and so we omit it. We remark that we know of no "direct" proof of Theorem 8.20.

In view of Richter's and Knight's results, a possible suggestion would be that for all n, if A has n-degree $\mathbf{a}$ then $\mathbf{a} = \mathbf{0}^{(n)}$. This is false. It is reasonable that it should be false since the coding techniques we have used need two jumps to decode. Ash, Downey, Jockusch and Knight proved the following definitive result:

Theorem 8.21 *(i)* **(Ash, Jockusch, Knight [8])** *If $n \geq 2$ is any computable ordinal and $\mathbf{a}$ is any degree with $\mathbf{a} > \mathbf{0}^{(n)}$ then there is a linear ordering A with proper n-degree $\mathbf{a}$.*

(ii) **(Downey and Knight [50])** *For any computable ordinal n there is a linear ordering A with proper n-degree $\mathbf{0}^{(n)}$.*

Some of the proofs in Theorem 8.21 use a new technique ("workers") which we will describe in the bulk of the rest of the section.

Nothing really new is needed for finite n (although worker arguments were used in (eg) Ash, Jockusch and Knight [8]). Suppose n is finite. We consider (ii) first. Again we use the Downey-Knight result Theorem 3.16. Indeed, this is what it was designed for. We already have $n = 1$ and $n = 2$. Use Theorem 8.7 (or in this case 8.2) in relativized form to get a linear ordering $D(= D(X)$ low ever X, but not isomorphic to any X=presentable linear ordering (and such that if Q is any X'-presentable linear ordering isomorphic to D then there is an X'-presentable linear ordering $R \cong Q$ such that $Q \not\leq_T R$, if we use Theorem 8.7). Now apply this to $X = \emptyset'$ via Theorem 3.16. Let $P = \gamma(D) = D(\eta + 2 + \eta)$. We claim P has proper 2-jump. Certainly the 2-jump degree of P is $\mathbf{0}''$. Now suppose that P has proper 1-jump degree $\mathbf{e}$. Then $\mathbf{0}' < \mathbf{e} < \deg(D)$. Let $E \cong P$ with $E' \in \mathbf{e}$. Then $E = \gamma(F)$, some $F \cong D$. Choose RrF with $R \;_T\!\not\geq F$. Then $\gamma(R)$ is isomorphic to P and has the wrong jump as $\gamma(R)' \leq_T R'$. (One can also use Knight's result to get this.) If P has 1st jump degree, it would need to be $\mathbf{0}'$ and so P would need a low copy E. Then $E = \gamma(F)$, some F of degree $\mathbf{0}'$.

Now, for arbitrary finite $n > 2$, we proceed as follows. We have done $n = 1$. For $n \geq 2$ choose D low over $\mathbf{0}^{(n)}$ not isomorphic to a $\mathbf{0}^{(n)}$ presentable ordering and having the additional property of Theorem 8.7. Now we know there is a copy of $\gamma(D) <_T \mathbf{0}^{(n)}$. Applying the γ operator via Theorem 3.16 n times we get $\gamma^n(D) <_T \mathbf{0}'$. Claim $\gamma^n(D)$ has proper $n+1$-degree $\mathbf{0}^{(n+1)}$. Clearly it has $n+1$-degree, say $\mathbf{a}$. Then $\mathbf{a} \leq \deg(D)$. Applying Theorem 8.7 we get some B with $B \cong A$ and $\deg(B) \not\geq \mathbf{a}$. Then $\gamma^n(B)$ does not have the right n-th jump.

To get (i) for $\mathbf{a} > \mathbf{0}^{(2)}$, we really need to code some invariant set into the ordering. For instance, we might attempt to do the following. Let $\mathbf{a} > \mathbf{0}^{(2)}$ choose Y with $Y'' \in \mathbf{a}$ and choose linear ordering A with the Theorem 8.7 property and low over Y'. We'd like then to argue that $\gamma(A)$ has proper 2-degree Y''. There seems no clear way to show this without more control over (eg) A, although it may be true. The problem is arguing that Y'' must be coded into all isomorphic copies of $\gamma(A)$. To overcome this we work with new orderings where we have more control over their order type and more control over an encoded set (to keep the n-jump up). Much of the machinery can be supplied from earlier sections.

For finite n, to get (i), since we are dealing with degrees $\mathbf{a} > \mathbf{0}^{(n)}$, we need a little care as to exactly which set we code into the ordering to code $\mathbf{a}$. We will use Lerman's [130] result. Recall that this involved $\tau(S)$ where if $S = \{a_0 < a_1 < \cdots\}$ then

$$\tau(S) = a_0 + \zeta + a_1 + \zeta + \cdots$$

and from Lerman's proof, it follows that $Dg(\tau(S)) = \{\deg(D) : S \text{ is computably enumerable in } D''\}$ where $Dg(X) = \{\deg(D) : D \cong X\}$ for a nontrivial linear ordering X. To get an ordering with arbitrary 2-degree let $X \;_T\!> \emptyset''$ then, via jump inversion, choose D with $D'' \equiv_T X$. Let $A = \tau(X \oplus \overline{X})$ be of degree D. It is easy to see that A has 2-degree $\deg(X)$ by Lerman's result, and cannot have 1st or 2nd jump degree by the Richter-Knight results.

We remark that using an infinite injury argument, Ash, Jockusch and Knight used a different ordering to code sets. This is of independent interest and we state their result (whose proof is similar to others we have seen) below.

Theorem 8.22 (Ash, Jockusch, Knight [8]) *Define $\sigma(R)$ to be the "shuffle sum" of R. That is, $\sigma(R)$ is composed of densely many copies of ω and $n+1$ for $n \in R$. Then for any R,*

$$Dg(\sigma(R)) = \{\deg(D) : R \textit{ is computably enumerable in } D''\}.$$

Clearly we could have use $\sigma(R)$ in place of $\tau(R)$ in the above reasoning. We also need four other lemmas whose proofs are quite technical and which we omit.

Lemma 8.23 (Macintyre [137]) *For any computable ordinal s and any A with $\emptyset^{(\alpha)} \leq_T A$ there is an α-generic set R with $R \oplus \emptyset^{(\alpha)} \equiv_T R^{(\alpha)} \equiv_T A$. (For our purposes, it is not really necessary to know what an α-generic set is, we only need the properties stated in Theorems 8.24 and 8.25 below.)*

Theorem 8.24 (Ash, Jockusch, Knight [8]) *If R is $\alpha + 1$-generic and $C = \{D : R$ is computably enumerable in $D^{(\alpha)}\}$ then for any $B \leq_T D^{(\alpha)}$, for all $D \in C$, we have $B \leq_T D^{(\alpha)}$. It follows that $\{D^{(\alpha)} : D \in C\}$ has no element of least degree.*

Theorem 8.25 (Ash, Jockusch, Knight [8]) *Let $C = \{D : R \leq_T D^{(\alpha+1)}\}$ for some R. If $B \leq_T D^{(\alpha)}$ for all $D \in C$ then $B \leq_T \emptyset^{(\alpha)}$. Thus if R is not computable in $\emptyset^{(\alpha+1)}$ then $\{D^{(\alpha)} : D \in C\}$ has no element of least degree.*

Finally, we will need one lemma when we deal specifically with limit ordinals.

Lemma 8.26 (Ash, Jockusch, Knight [8]) *If α is a computable limit ordinal, let $\{\alpha_n\}_{n\in\mathbb{N}}$ be a fundamental sequence of α. Let $R \subseteq \mathbb{N} \times \mathbb{N}$, with $R_n = \{k : (n, k) \in R\}$. Let $C = \{D : R_n \leq_T D^{(\alpha_n)}$ uniformly in $n\}$. Suppose that $B \leq_T D^{(\beta)}$ for all $D \in C$. If $\alpha_k > \beta$ and $T_k = \oplus_{n<k} S_n$ then $B \leq_T T_k^{(\beta)}$. Hence, if $\beta < \alpha_k$ and S_k is not computable $T_k^{(\alpha_k)}$, then $\{D^{(\beta)} : D \in C\}$ has no element of least degree.*

To get a linear ordering with proper 3-jump $\mathbf{a} > \mathbf{0}^3$, we can proceed as follows. Let R be 3-generic with $R^{(3)} \equiv_T \emptyset^{(3)} \oplus R$ where R^3 has degree $\mathbf{a}$. Now, via Theorem 8.24, if $C = \{D : R$ is computably enumerable in $D''\}$ then $\{D'' : D \in C\}$ has no element of least degree. Let $A_3 = \tau(S)$. Then $R \leq_T D^{(3)}$ for all $D \in C$ and as $R^{(3)} \equiv_T \emptyset^{(3)} \oplus R$, $R^{(3)} \leq_T D^{(3)}$ for all $D \in C$. It follows that A_3 has proper 3-degree.

For finite n we note the uniformities of the proof of Theorem 3.16 give

Lemma 8.27 *For any R and $n \leq \omega$*

(i) $Dg(\gamma^n(\tau(R))) = \{\deg(D) :$ *R is computably enumerable in* $D^{(n+2)}\}$

(ii) $Dg(\gamma^n(\tau(R \oplus \overline{R}))) = \{\deg(D) : R \leq_T D^{(n+2)}\}$

We remark that in Ash, Jockusch and Knight [8], they noted a similar generlization of Watnick's theorem (cf Roy and Watnik [199]). For instance in (i) we'd get $Dg(\zeta^n(\tau(R))) = \{\deg(D) : R$ is computably enumerable in $D^{(2n+2)}\}$. Using Lemma 8.27 and the facts on n-generic sets, we get (ii) for all finite n:

Lemma 8.28 *For each n with $2 \leq n < \omega$ and $\mathbf{d} > \mathbf{0}^n$ there is an ordering $A_n = A_n(\mathbf{d})$ with proper n-degrees $\mathbf{d}$.*

Proof For $n = m + 2$, let R have degree $\mathbf{d}$ and let D_0 be n-generic with $D^{(n)} \equiv_T R$. Let $A_n = \gamma^n(\tau(R \oplus \overline{R}))$. Let $C = \{D : R \leq_T D^{(n)}\}$. By Lemma 8.27(ii), $Dg(A_n) = \{\deg(D) : D \in C\}$, and R is the element of least degree in C. Finally A_n does not have $n-1$ degree by Theorem 8.25. $\square$

When we proceed to infinite cases we run into a problem with the above. The point is now that encoding a set as above is no longer a classical invariant. We need a new idea. For the case of limit ordinals the idea is to piece together smaller pieces in some fashion, and it is here we use Lemma 8.26. For instance, let $\alpha = \omega$, let $R \subseteq \mathbb{N} \times \mathbb{N}$ and $R_n = \{k : \langle n, k\rangle \in R\}$. Now let $\lambda(R) = \sum_{n<\omega} \gamma^n \tau(R_n)$. It is not difficult to see that the uniformities of the previous work show:

Lemma 8.29 *$Dg(\lambda(R)) = \{\deg(D) :\ R_n$ is computably enumerable in $D^{(n+2)}$ uniformly in $n\}$.*

We remark that Ash et al [8] used $\rho(R) = \sum_{n<\omega} \zeta^n \sigma(R_n)$ where

Lemma 8.30 (Ask, Jockusch, Knight [8]) *$Dg(\rho(R)) = \{\deg(D) :\ R_n$ is computably enumerable in $D^{(2n+2)}$ uniformly in $n\}$.*

It is not hard to apply either Lemma 8.30 or 8.29 to any $\mathbf{d} > \mathbf{0}^{(\omega)}$ via Macintyre's result and Lemma 8.26. That is,

Theorem 8.31 (Ask, Jockusch, Knight [8]) *If $\mathbf{d} > \mathbf{0}^{(\omega)}$ there is an ordering A_ω with proper ω-degree $\mathbf{d}$.*

Proof Choose ω-generic R with $R^{(\omega)} \equiv_T \emptyset^{(\omega)} \oplus R$ and $R^{(\omega)}$ of degree $\mathbf{d}$. We know for all m, k that $R_{k+1} \leq_T (R_0 \oplus \cdots \oplus R_k)^{(m)}$. Let $A_\omega = \lambda(R)$. If $C = \{D : R_k$ is computably enumerable in $D^{(2+k)}$ uniformly in $k\}$ then by Lemma 8.26 $Dg(A_\omega) = \{\deg(D) : D \in C\}$. So A_ω has ω-jump $\mathbf{d}$. To see that it is proper, suppose it had n-degree for some n. Then there is some $D_0 \in C$ with $D_0^{(n)} \leq_T D^{(n)}$ for all $D \in C$. It follows, by Lemma 8.26, that if $k+2 > n$ then $D_0^{(n)} \leq_T (R_0 \oplus \cdots \oplus R_{k-1})^{k+2}$ uniformly in k. Hence $D_0^{(k+2)} \leq_T (R_0 \oplus \cdots \oplus R_{k-1})^{(m)}$, some m, and hence $R_k \leq_T D_0^{(k+3)}$ so that $R_k \leq_T (R_0 \oplus \cdots \oplus R_{k-1})^{(m+1)}$, a contradiction. $\square$

For $\alpha > \omega$ we really need something new. This is where the "worker" idea comes in. We will look in detail at the case $\alpha = \omega + 1$ (which gives the main idea) and only give some remarks on how one treats the general case.

First we wish to give the reader some intuition as to the "methods of workers" first introduced by Harrington [86] [see also Calhoun [25, 26] for a more formal presentation of these ideas]. Consider the typical way we describe a priority argument. We tend to isolate the particular requirements and then discuss how the strategies cohere. This usually involves discussion of the true path (where we really would like to work) but then we must also discuss how the *computable version* of the construction works, where we only have approximations to the true path.

For instance, consider a typical injury priority argument. We call this more precisely a $\emptyset'$-priority argument since we know that $\emptyset'$ can sort out exactly how we meet the requirements. Consider the result Theorem 3.16 that R is X'-presentable iff $\gamma(R)$ is X-presentable. Let $X = \emptyset$ for the unrelativised version. Let $R \subseteq \mathbb{Q}$. For each $x \in R$ we wait to replace it with a successivity and put η's between the successivity. If R were computable, this would pose no problem. In the worker model we have two workers, worker 1 and worker 0. Worker 1 works at the $\emptyset'$ level. It knows everything about R. Its task is to enumerate R as a subset of $\mathbb{Q}$ and tell worker 0 to make the $x \in R$ into successivities and put η's between them. But, as often happens between management and the floor staff, worker 0 has difficulty understanding worker 1's commands. Worker 0 only lies at the computable level. By the limit lemma, worker 0 can only comprehend worker 1's command via a series of Δ_2 approximations. That is worker 0 has at its disposal a computable $f(x, s)$ where for all x, $\lim_s f(x, s) = R(x)$. So for x, worker 1 says either $x \in R$ or $x \notin R$. Worker 1 only gets $f(x, 0),\ f(x, 1),\ f(x, 2), \ldots$. However, it knows that eventually we reach a t such that $f(x, t')$ is constant for all $t' > t$. The traditional "coherence" strategy of a priority argument can be paraphrased by saying that worker 0 can live with a finite number of mistakes. Of course we should always really represent $\overrightarrow{f(x, s)}$ i.e. $f(0, s), \ \ldots, \ f(x, s)$ in place of $f(x, s)$ so we know the values *together.* This is important in the later more complex arguments.

Let us now consider a typical $\emptyset''$ argument. Since we have already a proof of it, let us again pick on Watnick's theorem. So now R is $\emptyset''$ presented and we want ζR to be computably presentable. For such an argument we have 3 workers. Worker 2 works at the $\emptyset''$ level, 1 at the $\emptyset'$ level and 0 at the computable level. Worker 2 knows all of R, enumerates its points, issues commands such as "build ζ around each point".

Worker 1 can see worker 2's commands via the limit lemma. This worker uses $f(x, s)$ with $\lim_s f(x, s) = R(x)$ existing, f computable in $0'$ and when $f(x, s) = 1$ it says to worker 0 put ζ around x. When $f(x, s) = 0$ it says to worker 0, don't put ζ around x, and fix up your mistakes if $f(x, s - 1) = 1$.

At the bottom level worker 0 receives 1's commands via the limit lemma, but now things are more complex. We know that f behaves such as, perhaps

$$f(x, 0) = 0, \quad f(x, 1) = 1, \quad f(x, 2) = 0, \quad f(x, 3) = 0, \quad \cdots .$$

Now g, which approximates 1's commands, needs to approximate the *binary* function f. So g can be wrong finitely often on each (x, i). So typically $f(x, i) = j$ corresponds to a finite number of $g(x, i, \cdot)$ values. For instance, $g(x, i, 0),\ g(x, i, 1),\ \ldots,\ g(x, i, s) = j$ and then $g(x, i, s) = j$ thereafter.

This is why worker 0 only sees the correct value of $R(x)$ infinitely often.

Let us now see in detail how this actually works. One sleight-of-hand in the above presentation is that, at any stage s, we only deal with finite portions. So it is not correct to say worker 1 issues a command to build a ζ-piece around x. This is only "globally" true if the command is "really" true. (Note the use of the recursion theorem.)

At stage s, worker 2 computes $R_s \subseteq \{q_0, \ \ldots, \ q_s\}$ with q_i the ith member of $\mathbb{Q}$. We will replace the explicit use of the limit lemma by the expression "guesses". That is if $f(x, s) = 1$, as above, then worker 1 guesses $x \in R$.

Worker 1 guesses R_s. Let k_s be the first k with $k = k_{s-1} + 1$ or k the first place with the guess changing from $s - 1$. The worker issues an instruction i_s via k_s. This either says destroy some previous work, or begin to build a ζ interval around k_s. The latter occurs if k_s now appears to be in R and the former if k_s was once in R but now is guessed out of R (i.e. currently appears to be out of R). These must cohere in the sense that we can carry out i_s in the environment left by $i_0, \ldots, i_{s-1}$. That is, *we need to be able to correct mistakes.* By our analysis of Watnick's theorem in Section 5, we know how to destroy old work coherently: we incorporate it into other blocks. In the end, worker 1 produces an eventually stable set of commands $\{i_{j_0}, i_{j_1}, \ldots\}$ which are not subsequently revoked and replaced with contradictory commands.

Worker 0 just works one level down. The way this is carried out is discussed in detail in Section 5 so we omit translation into this language. The crucial thing to keep in mind is that everybody needs to be able to be sufficiently conservative so that its *stable* set of actions (instructions) does the correct thing and the false commands or actions can be repaired.

So what is the point of this? First it gives another viewpoint as to how priority arguments work and a flexible model for constructions. At this stage, we should mention that there are several other formalisations of the idea of a "depth-α" priority argument. One can use iterated trees of strategy (Slaman, and especially Lempp-Lerman [129]). Another approach via iterated trees is formalised in the work of Ash on *α-systems* (Ash[4, 5, 6, 7]). In any construction, of course, all we "really do" is approximate a $\mathbf{0}^{(\alpha)}$ predicate computably. (In some sense this idea is used in Feiner [64].) But this is an inadequate description since the crucial thing is to say how to organise the strategies to be able to get the requirements to cohere. This is why, for instance, the tree approach is often nicer than others since we put the "atomic outcomes" on the tree and can "see" what is happening. However, in many inductive instances, it is undesirable perhaps to use all the tree machinery and sometimes the worker approach seems the most perspicuous. Finally there have been a number of attempts to prove metatheorems for these depth n arguments. In particular, first Ash [4] (with α-systems) and later Knight [120, 121] developed rather general conditions under which one can construct a structure B obeying the appropriate list of conditions. The problem with these (in the opinion of the author) is that it often seems more difficult to verify the conditions than to perform a direct construction. The final verdict will be determined by what is the more useful.

Using arguments with infinitely many workers, Ash, Jockusch and Knight proved Theorem 8.21(ii) as follows. Let α be a computable ordinal. Let $\nu^\alpha(R)$ be the sum of densely many copies of $q+1+\sum_{1\leq\alpha<\beta}\zeta^\beta$ for $q \in \omega$, $\quad i \leq \beta < \alpha$ and $r+1+\sum_{1\leq\gamma<\alpha}\zeta^\beta$ for $r \in R$. Given a fundamental sequence $(\alpha_n)_{n\in\omega}$ for α, define a new sequence $(\alpha_n^*)_{n\in\mathbb{N}}$ via

(i) if $|\alpha_n| < \infty$, let α_n^* be the first $k \geq 3$ with $k \geq \alpha_n$ and if $n > 0$ then $k > \alpha_{n-1}$,

(ii) if $|\alpha_n| = \infty$ and β is the greatest limit ordinal with $\beta \leq \alpha_n$, $\mu = \beta + k$ for some odd integer k, and if $n > 0$ then $\mu > \alpha_{n-1}$.

Definition 8.32 (Ash, Jockusch,Knight [8]) *Let α be a computable ordinal and $R \subseteq \omega$. Define $A_\alpha(R)$ via*

(i) if $\alpha = 2m+2 < \infty$ then $A_\alpha(R) = \zeta^m\sigma(R \oplus \overline{R})$.

(ii) If $\alpha = 2m+3 < \infty$ then $A_\alpha(R) = \zeta^m\sigma(R)$.

(iii) If α is a limit ordinal and (α_n^*) is defined as above, then

$$A_\alpha(R) = \Sigma_{n\in\omega}(1+\eta+1+A_{\alpha_n^*}(R_n =_{\mathrm{def}} R \cap \omega^{(n)})).$$

(iv) If β is a limit ordinal and $\alpha = \beta+1$ then $A_\alpha(R) = \nu^\alpha(R \oplus \overline{R})$.

(v) If β is a limit ordinal and $\alpha = \beta + 2$ then $A_\alpha(R) = \nu^\alpha(R)$.

(vi) If β us a limit ordinal and $\alpha = \beta + 1 + 2k + 2$ then let $A_\alpha(R) = \zeta^{\beta+k}\sigma(R \oplus \overline{R})$.

(vii) If β is a limit ordinal and $\alpha = \beta + 2k + 4$ then let $A_\alpha(R) = \zeta^{\beta+k}\sigma(R)$.

Lemma 8.33 (Ash, Jockusch, Knight [8]) *Let $A_\alpha = A_\alpha(R)$*

(i) For finite even $\alpha = 2m + 2$, $Dg(A_\alpha) = \{\deg(D) : R \leq_T D^{(\alpha)}\}$.

(ii) For finite odd $\alpha = 2m + 3$, $Dg(A_\alpha) = \{\deg(D) : R$ is computably enumerable in $D^{(\alpha)}\}$.

(iii)

(iv) For α limit, $Dg(A_\alpha) = \{\deg(D) : R_n \leq_T D_n(\alpha_n^*)$ uniformly in $n\}$.

(v) For $\alpha = \beta + 2k + 1$ with β limit and $k \in \omega$, $Dg(A_\alpha) = \{\deg(D) : R \leq_T D^{(\alpha)}\}$.

(vi) For $\alpha = \beta + 2k + 2$ with β limit and $k \in \omega$, $Dg(A_\alpha) = \{\deg(D) : R$ is computably enumerable in $D^{(\alpha)}\}$.

Applying Lemma 8.33 to an α-generic R via Theorems 8.24, 8.25 and Lemma 8.26, we get (in the same way as for finite n) Theorem 7.26(i). We will not give the detailed proofs associated with Lemma 8.33. Rather we will briefly outline an ω-worker proof of a result from the Ash et al [8] paper:

8.34 *If $D \leq_T \mathbf{0}^{(\omega+1)}$ then $\zeta^{(\omega)}D$ is computably presentable.*

Proof sketch The proof of this infinite generlization of Watnick's theorem uses one worker for each $n < \omega$ and two limit workers: ω and $\omega + 1$. Define the relation $\sim_n$ as follows: Set $x \sim_0 y$ if there are only finitely many elements between x and y, $s \sim_{\alpha+1} y$ if there are only finitely many α-equivalence classes between $[x]_\alpha$ and $[y]_\alpha$, and for limit β, $x \sim_\beta y$ if for some $\alpha < \beta$, $x \sim_\alpha y$.

Now worker $\omega + 1$ enumerates the ordering D. Worker ω is given an approximation to D. At stage s, this is those e such that for some $C \leq s$, C is a halting computation of $\Phi_e^{\emptyset^{(\omega)}}(e)$, in which there are no questions asked of $\emptyset^{(n)}$ for $n > s$. Let $\emptyset_s$ be this approximation. We see that $e \in \emptyset^{(\omega+1)}$ iff $e \in \emptyset_s$ for all sufficiently large s. The point is that worker ω can try to use $\emptyset_s$ instead of $\emptyset^{(\omega+1)}$ (to compute D via $\Gamma(\emptyset^{(\omega+1)}) = D$) and bounding computations by s. Then if we let $f^*(n, s)$ be the guess to $D(n)$ obtained this way, since $f^*(k, n)$ is uniformly computable in $\mathbf{0}^{(n)}$, we can let worker n operate at the $\mathbf{0}^{(n)}$ level and then worker n can compute $f^*(k, n)$ in the same way that worker ω can.

Now the strategy is rather straightforward; the Watnick strategy can be fully approximated so that worker $n+2$ can issue commands to workers $n+1$ and n in exactly the same way, but using the approximation given as above. Then worker $n + 2$ will succeed in commanding worker 0 to build ζ^n blocks. Finally the only difference is that when we correct at worker n then we correct via $\sim_n$ in place of $\sim_1$ and $\sim_0$ in Watnick's theorem. One then argues by induction, that for all n, worker n's commands are carried out. $\square$

The actual details of these arguments seem very messy, and certainly in many applications the details are given in a rather informal way (but see Calhoun [25, 26]). The work of Ash and of Lempp-Lerman gives formal versions of arguments along similar lines.

For instance, Ash viewed the arguments as examples of metatheorems. Here is one example.

Let U and L be computably enumerable sets and P a computably enumerable tree consisting of finite nontrivial sequences $\ell_0 u_1 \ell_1 u_2 ...$, with ℓ_0 and $\ell_n, u_{n+1} \in L \times U$. Let E be a computable partial function called the enumeration function, with $E : L \mapsto \{F \subset \mathbb{N} : F \text{ finite}\}$. We let q be a Δ^0_α function from the members of P of odd length (i.e. ending in L) such that if $q(\sigma) = u$ then $\sigma u \in P$. The function q is called the instruction function and corresponds to the α level worker in the worker argument. A path $\pi = \ell_0 u_1 \ell_1 u_2 ...$ of P is called a *run* if ot obeys the instruction function. That is, for all n $q(\ell_0 ... \ell_n) = n_{n+1}$.

Ash then defined a *α-system* $\mathcal{A}$ to be a structure $(L, U, \ell_0, P, E, \leq_\beta : \beta \leq \alpha)$ as above for which additionally,

(1) for all $\beta \leq_\beta$ is transitive and reflexive,

(2) if $\gamma < \beta$ then $\ell \leq_\beta \widehat{\ell}$ implies $\ell \leq_\gamma \widehat{\ell}$,

(3) if $\ell \leq_0 \widehat{\ell}$ then $E(\ell) \subseteq E(\widehat{\ell})$,

(4) if $\sigma u \in P$, σ ends in ℓ^0, $\ell^0 \leq \beta_0 \leq \ell^1 \leq_{\beta_1} \cdots \leq \beta_{k-1} \ell^k$, and $\alpha > \beta_0 > \cdots > \beta_k$, then there is an m such that $\sigma u m \in P$ and $\ell^i \leq_{\beta_i} m$.

Theorem 8.35 (Ash [3, 6, 7]) *Given an α-system $\mathcal{A}$, and q a Δ^0_α instructio function for P there is a Δ^0_α run π of (P, q) such that $E(\pi)$ is computably enumerable.*

The proof of the above works by induction for finite α and then along some system of notations for arbitary computable α. For an pplication yet again we look at Watnick's Theorem. So we suppose that $(A \leq_A)$ is a ordering of type τ with τ Δ^0_α computable. Let C be a computable set upon which we will construct the computable linear ordering. For suitable ordering $(X, \leq_X)$ of type β (e.g. we have $\alpha = 2n$ and $\beta = \zeta^n$, as usual), we define T to be the tree of all nonempty finite sequences $(B_0, f_0, B_1, f_1 ...)$ such that

(i) $B_i = (B_i \leq_i)$ is a linear ordering, with $B_i = \{a_1, ..., a_{k_i}\}$ for some k_i and $k_0 = 0$,

(ii) B_{i+1} extends B_i,

(iii) $f_i : C \mapsto X \times B_i$ is a partial injective function,

(iv) f_{i+1} extends f_i,

(v) dom f_{i+1} contains the first i elements of C, and

(vi) ran f_{i+1} contains the first i elements of $X \times A$ which are in $X \times B_i$.

To use Theorem 8.35, we arrange that the ordering on C induced by $f = \cup_n f_n$ is computable, or equivalently computably enumerable. We define $E(B_0 f_0, ..., B_n f_n)$ to be the finite ordering induced on C by f_n; $\{(p, q) : p, q \in \text{dom} f_n \wedge f_n(p) \leq f_n(q) \in X \cdot A\}$. To then apply the metatheorem we need suitable relations $\leq_\beta$.

This is achieved via so-called β maps, where we say that $g : \mathcal{A} \mapsto \mathcal{B}$ is a β map if, for dom $g = \{a_1, ..., a_n\}$, the every Π_β sentences true $(\mathcal{A}, a_1, ..., a_n)$ is also true in $(\mathcal{B}$. Then for $\sigma = (B_0, f_0, ..., B_m, f_m)$ and $\tau = (B_0, f'_0, ..., B_n, f'_n)$, define $\sigma \leq_\beta \tau$ iff dom $f_m \subseteq$ dom f'_n, and $f'_n f_m^{-1}$ is a β map from $X \cdot B_m$ to $X \cdot B_n$.

The above is a very brief sketch of Ash's ideas and we urge the reader to look at Ash [7] and the forthcoming monograph Ash-Knight [**?**] for details. One of the problems with this and the even more elaborate Lempp-Lerman pproach is that one ofthe must vary the metatheorem to get the detils to work in new settings, and even then work must be done do actually verify the conditions.

We mention one recent application where such a varition was needed. Knight [**?**] who proved, that the set of limit points of a orderng of type ω^2 can have arbitary 2REA degree.

In any case, we have dealt with degrees strictly above $\mathbf{0}^{(\alpha)}$. To finish the proof, we need to get orderings of proper degree $\mathbf{0}^{(\alpha)}$ for $\alpha \geq \omega$. These are obtained via similar arguments where one comes up with an order type to code the appropriate sufficiently generic set (in the correct way). These results are rather technical an along similar lines so we omit them (see Downey-Knight [50] for details). □

Before leaving this topic, we mention one last result.

Theorem 8.36 (Ash, Jockusch,Knight [8]) *There is an ordering that does not have α-degree for any computable α.*

Proof sketch In Ash et al [8], the authors claim that one can show, via the forcing introduced in Ash, Knight, Manasse and Slaman [10], that ω_1^{ck} has no α-degree. □

Question 8.37 *Under what conditions does an order type have an α-degree?*

One can use worker arguments for many other problems. One of the first applications to orderings as due to Ash [4, 5]. Here, Ash investigated conditions under which a computable model A was Δ_α *stable*. This means that for all computable $B \cong A$, $B \cong_\alpha A$. That is, *all* isomorphisms between B and A are Δ_α. This is not to be confused with Δ_α-*categorical*. This means that for any computable $B \cong A$ *there is* a Δ_α isomorphism between B on A. For instance $\mathbb{Q}$ is computably categorical yet is not Δ_α stable for any α (as there are uncountably many automorphisms of $\mathbb{Q}$). Now by our earlier results, we know that ω is not computably categorical, (and so not computably stable) yet it is easy to see that ω is Δ_2 stable. Recall that a computable structure A is *rigid* if it has no nontrivial automorphisms (Ash [4] observed that A is Δ_n stable iff for some set of constants $\{P_1, \ldots, P_n\} = P$, (A, P) is rigid and Δ_n-categorical). Ash used his α-system arguments to characterise Δ_α-stable structures in terms of certain computable infinitary formulae.

Barker [14] did the same for Δ_n-categoricity. These results were subject to certain decidability conditions, which Ash, Knight, Manasse and Slaman [10] observed could be eliminated if the results were asked to hold absolutely for all relativisations. These results can be applied to linear orderings to show, for instance, ω^α *is* $\Delta_{\alpha+1}$-*stable for all* α *for all computable* α (Ash [4, 5, 6, 7]).

As our final example of this technique we briefly outline a solution to a question of Rosenstein [195, 196] (also Lerman-Rosenstein [132]), using a worker argument.

Theorem 8.38 (Downey) *There is a computable, nonscattered linear ordering with no arithmetical dense subset.*

Proof sketch We build A of order type $\zeta^{\omega}\eta$. We must only ensure that if B is an infinite arithmetical subset of A then for some $x,\, y \in B,\; x \sim_n y$. We start with $q_0,\, q_1,\, \ldots$, a copy of $\mathbb{Q}$. We must define an isomorphism f with domain $\mathbb{Q}$ (i.e. ensure that $f(q_i)$ exists for all i) and have a level $2n$ task of building ζ around each $f(q_i)$. The only requirement we need add is that if A_i is the i-th arithmetical set and $|A_i| = \infty$ then for some $n,\; x \sim_n y$ for some $x,\, y \in A$. We can enumerate the arithmetical sets as $A_0,\; A_1,\; \ldots$ so that A_i is computable in $\mathbf{0}^{(i)}$. We let worker $2n+2$ diagonalise argument A_n. We do not let worker $2n+2$ change $f(q_i)$ for $i \leq 2n+2$. We let worker $2n+2$ act if it sees $x,\, y \in A_n$ with $x,\, y$ not in the same block as $f(q_i)$ for $i \leq 2n+2$, and such that if we declare $x \sim_{2n+2} y$ then we won't amalgamate $f(q_i)$ and $f(q_j)$ for $i,\, j \leq 2n+2$. Worker $2n+2$ then asks that $x \sim_{2n+2} y$. We must then build a new copy of some $f(q_j)$ for some $j \geq 2n+2$. For instance if $x \sim f(q_j)$ and $y \sim f(q_k)$ then we might need to build a new $f(q_k)$ (eg, if $k > j$). We can fix up errors by asking that if we begin a new $f(q_k)$ and later (eg) $y \in A_n$, then we cancel this (forever) and go back to the old one. The details go through in the standard way. $\square$

We close this treatment with a question suggested by Theorem 8.38.

Question 8.39 *Is there a computable non-scattered linear ordering not isomorphic to one with arithmetical dense subset?*

9 Other Work and Related Results

In this section we will look briefly at some other related work. Many of our arguments are diagonalisations. Jockusch and Soare [101] asked if there is an analogue of the recursion theorem for orderings. Let

$$L = \{e : \varphi_e \text{ is the characteristic function of a linear of } \omega\}$$

and let A_e denote $\langle \omega, \varphi_e \rangle$ if $e \in L$. Jockusch and Soare showed

Theorem 9.1 (Jockusch and Soare [101]) *There is a computable f such that for all $e \in L$, $f(e) \in L$ and $A_{f(e)} \ncong A_e$.*

They point out that the following analogue of the recursion theorem is, however, open.

Question 9.2 (Jockusch and Soare [101]) *Is there a computable f with the following properties:*

(i) $(\forall e)[e \in L \to f(e) \in L]$

(ii) $(\forall e \in L)(A_{f(e)} \ncong A_e)$

(iii) f is well defined on order types. That is $(\forall a, b)[(a, b \in L$ *and* $A_a \cong A_b) \to A_{f(a)} \cong A_{f(b)}]$.

There is a lot of interesting related work on Boolean algebras. Remmel [186] is an excellent source here. Many analogous questions seem much harder for Boolean algebras since so many order types correspond to the same Boolean algebra, but things can differ. Downey and Jockusch [49] proved that every low boolean is isomorphic to a computable one. John Thurber demonstrated that each low_2 boolean algebra is also iomorphic to a computable one.

Question 9.3 *Develop results akin to those of Ash, Downey, Jockusch and Knight for boolean algebras. In particular, is there a low_3 boolean algebra not isomorphic to a computable one? Also what is the precise relationship between orderings and associated algebras? Finally, what is the effective content of the Ketonen invariants ([104]) classifying countable boolean algebras?*

There is some work here. Jockusch and Soare have proven that no boolean algebra can have proper finite n-th jump degree $\mathbf{a} > \mathbf{0^{(n)}}$. But any degree $\mathbf{a}$ above $\mathbf{0}^\omega$ can be the proper ω-th jump degree of a boolean algebra. We refer the reader to Downey [**?**] for more on this topic.

Feiner [63, 64] showed that there were Π_1 boolean algebras not isomorphic to computable ones. Thus the argument will be much more complicated than that used for orderings. Feiner's result on Π_1 boolean algebras in some ways is a forerunner to the worker arguments we have seen in Section 8 and was a very fine achievement for the 1960's.

There are of course many other ordered structures we have not examined. For instance, Alex Feldman [61] has some interesting results on recursion theoretical aspects of a lower semilattice (rather along the lines of the Metakides-Nerode-Remmel approach of looking at lattices of substructures).

Another area we have not looked at is ordered structures, Metakides and Nerode [150] analysed Craven's [32] classification of the cone of orderings of a formally real field. They showed that for any Π^0_1 class P, there is a computable formally real field whose orderings are in one-one correspondence with the members of P (in particular up to Turing degree). Thus, for instance,

Theorem 9.4 (Metakides and Nerode [150]) *There are computable formally real fields with no computable orderings and even some with all their orderings of incomparable Turing degrees.*

Question 9.5 *If A is a formally real computable field, is A isomorphic to a computably ordered computable formally real field?*

Downey and Kurtz [51] looked at the analogous result for ordered abelian groups. The question is nontrivial since it is not clear when the fact that a field A can be ordered as an additive group implies that it can be ordered as a field.

Theorem 9.6 (Downey and Kurtz [51]) *There is a torsion free computable abelian group G with no computable orderings. Indeed G has infinite rank yet its only computably orderable subgroups have finite rank.*

Question 9.7 (Downey and Kurtz [51]) *(i) Can one construct a computable G whose orderings have exactly the degrees of a given Π^0_1 class?*

(ii) Is every orderable computable (abelian) group isomorphic to a computably orderable computable group?

We remark that little is known about orderings of computable nonabelian groups. For instance, by a classical result of B. Neumann [166] any free group is orderable. Also by N. Neumann [167], every ordered group is isomorphic to a quotient of (an ordering of) a free group by a convex normal subgroup. Despite the highly noneffective classical proof, recently Reed Solomon has proven that there is a computable version of this result. However, many questions suggest themselves.

Question 9.8 *What is the effective content of Hölder's theorem [92] on Archimedean ordered groups?*

We refer to Solomon [**?**] for more details and for a number of very attractive results on computable ordered groups.

Another area that has attracted a great deal of attention recently is *subcomputable* or *feasible* algebra. Here one studies not computable, but polynomial time, analogues of classical objects. So a linear ordering L is polynomial time presented if the relation "$z \in L$" is p-time in $|x|$. Consider $x \in \{0, 1\}^{<\omega}$. Cenzer and Remmel [27], Grigorieff [82], Nerode and Remmel [165] and others have obtained very interesting results here. We will satisfy ourselves by merely quoting two results.

Theorem 9.9 *(i)* **(Cenzer and Remmel [27])** *There is a computable linear ordering of type $\omega + \omega$ not computably isomorphic to a p-time ordering.*

(ii) **(Remmel (unpublished))** *In (i) we can replace $\omega + \omega$ by $\omega + \zeta\lambda + \omega^*$.*

Theorem 9.10 (Grigorieff [82]) *Every computable linear ordering has a copy in* $\mathrm{DTIME} - \mathrm{SPACE}(n, \log(n))$.

This is obviously an area of great potential.

The final one we will mention is an old one that was part of a major area of research in the 1960's: *constructive order types.* Dekker and Myhill asked what happens if we can only see comparability relations through "computable eyes". Suppose we have a partial computable injective function f. Evidently from an effective point of view dom f and ra f are the "same size". But clearly f matches many other sets. For if $A \subseteq \mathrm{dom} f$ then A is matched to $f(A)$. This leads to the following definition: we say that Q is *computably equivalent* to R ($Q \approx R$) if there is an injective partial computable function f with $\mathrm{dom} f \supset Q$, $\mathrm{ra} f \supset R$ and $f(Q) = R$. It is easy to see that $\approx$ is an equivalence relation and the equivalence classes are called *computable equivalence types* (CET's). If A is a set, let (A) denote the CET of A. CET's have been extensively studied by Dekker, Myhill, Barback, Mclaughlin, Nerode and others. The reader is directed to Mclaughlin [**?**] for details of this development. One thinks of computable equivalence as generating an analogue of the equipollence. We say $(A) \leq (B)$ iff there are representatives of A and B of (A) and (B) and a set C with $A \oplus C \approx B$. One of the key notions is that of an *isol.* We say that (A) is an *isol* if A is finite or immune. The isols are the essential building blocks of this system since *A is an isol iff A is not computably equivalent to a proper subset of itself.* So isols are the analogues of the Dedekind finite sets.

The major idea in the programme of investigating CET's was to try to understand what arithmetic truths of the integers would smoothly lift to the isols. For instance, we define $+$ and $\cdot$ in the obvious ways, and soon discover that for isols many classical combinatorial identities are true for the isols. This work is particularly interesting in view of the fact that we now know that the isols really are models of *choice free mathematics obtained via Kleene realizablity* (cf McCarty [144]). Thus they are really strongly related to classical intuitionistic systems. In many ways, the culmination of much of the early work on isols were the beautiful results of Nerode [163] who extended some earlier work of Myhill to show how to lift a wide class (Horn sentences) of properties to the isols. Viewed nowadays, Nerode's argument is clearly a forcing construction (before Cohen, although not set theoretical forcing) and later Ellentruck [60] recast Nerode's work in terms of boolean valued models. So the CET work can be viewed as looking at preservation theorems for a certain class of forcing construction. The reader should look at Crossley [36] for a nice amount of the development and philosophy of some of the originators of the area. It is quite interesting to speculate on whether the methods or ideas can be used to capture a fragment of uniform (classical) combinatorics.

Crossley [33] and Manaster (unpublished) both began to look at similar analogues of the ordinals and were led to several definitions of "effective order type" (also Rice [187]). We look at only one (see eg Eisenberg and Remmel [58] for others). We say $A \approx B$ (A and B have the *same constructive order type*) if $A, B \subseteq \mathbb{Q}$ and there is an injective partial computable f with $f(A) = B$ and such that if $f(x)\downarrow$ and $f(y)\downarrow$ then $x \leq y$. Again we let (A) denote the equivalence class, the COT, of A. An isolated COT is called *Dedekind.* Crossley [33] developed quite an elaborate theory of COT's and Dedekind COT's. For instance, he showed that

Theorem 9.11 (Crossley [33]) *If P is a unary nontrivial function built up from $+$, $\cdot$, expo-*

nentiation, taking converses and allowing Dedekind COT parameters, then for Dedekind COT's

$$P(X) = P(Y) \quad \leftrightarrow \quad X = Y.$$

Crossley and Nerode [37] later pushed Theorem 9.11 a lot further (along the lines of Nerode [163]), and to much more general settings. Recently Nerode and Remmel [165] have looked at p-time versions of this work.

Finally one area we have not even looked at is *computable graph theory.* Here there is an extensive literature such as Manaster and Rosenstein's [140] computable analysis of P. Hall's marriage theorem. Other work can be found in, for instance, Aharoni, Magidor and Shore [1], Bean [15, 16], Biegel and Gasarch [17, 18, 19, 20], Gasarch [73], Gasarch, Kueker and Mount [76] Gasarch and Lockwood [74], Hirst [91], Jockusch [98], Kierstead [111, 112], Remmel [183, 184], Schmerl [215, 216, 217], Shore [**?**] and Tverberg [222]. Here we refer to Gasarch's survey in the present volume.

References

[1] Aharoni, R., M. Magidor, and R. Shore, On the strength of König's duality theorem for infinite bipartite graphs, Journal of Combinatorial Theory, B. vol. 54 (1992), 257-290.

[2] Anderson, I., *Combinatorics of Finite Sets* Oxford Science Publ., Clarendon Press, Oxford, 1987.

[3] Ash, C. J., Stability of recursive structures in the arithmetical degrees, Ann. Pure and Appl. Logic, vol. 32 (1986), 113–135

[4] Ash, C. J., Recursive labelling systems and stability of structure in the hyperarithmetical degrees, Trans. Amer. Math. Soc., vol. 298 (1986), 497–514, correction (ibid) vol. 310 (1988), 851.

[5] Ash, C. J., Categoricity in the hyperarithmetical degrees, Ann. Pure and Appl. Logic, vol. 34 (1987), 1–34.

[6] Ash, C. J., Labelling systems and r.e. structures, Ann. Pure and Appl. Logic, vol. 47 (1990), 99–120.

[7] Ash, C.J., A construction for recursive linear orderings, J. Symb. Logic Vol. 56, (1991), 673-683.

[8] Ash, C. J., C. Jockusch and J. F. Knight, Jumps of orderings, Trans. Amer. Math. Soc. Vol. 319, (1990), 573-599.

[9] Ash, C. J. and J. F. Knight, Pairs of recursive structures, Ann. Pure and Appl. Logic, vol. 46 (1990), 211–234.

[10] Ash, C. J., J. F. Knight, M. Manasse, and T. Slaman, Generic copies of countable structures, Ann. Pure and Appl. Logic, vol. 42 (1989), 195–206.

[11] Ash, C. J. and A. Nerode, Intrinsically recursive relations, in Crossley [35], 26–41.

[12] Baumslag, G., E. Dyer, and C. Miller III, On the integral homology of finitely presented groups, Topology, Vol. 22 (1983), . 27-46.

[13] Baker. K., D. C. Fishburn and F. Roberts, Partially ordered sets of dimension 2, Networks, vol. 2 (1971), 11–28.

[14] Barker, E., Intrinsically Σ^0_0 relations, Ann. Pure and Appl. Logic, vol. 39 (1988), 105–130.

[15] Bean, D., Effective coloration, J. Symb. Logic, vol. 1 (1976), 469–480.

[16] Bean, D., Recursive Euler and Hamilton paths, Proc. Amer. Math. Soc., vol 55 (1976), 385–394.

[17] Biegel, R. and W. Gasarch, On the complexity of finding the chromatic number of a recursive graph I: The bounded case, Ann. Pure and Appl. Logic, vol 45 (1989), 1–38.

[18] Beigel, R. and W. Gasarch, On the complexity of finding the chromatic number of a recursive graph II: The unbounded case, Ann. Pure and Appl. Logic, vol. 45 (1989), 227–246.

[19] Beigel, R. and W. Gasarch, The mapmakers' dilemma, University of Maryland Technical Report TR 2262.

[20] Beigel, R. and W. Gasarch, Applications of binary search to recursive graph theory, University of Maryland Technical Report TR 1804.

[21] Bonnett, R., Stratification et extension des genres de chaines denombrables, CRAS (Paris), vol. 269 (1969), 880–882.

[22] Bonnet, R. and M. Pouzet, Extension et stratification d'ensembles disperses, CRAS (Paris), vol. 268 (1969), 1512–1515.

[23] Bonnet, R. and M. Pouzet, Linear extensions of ordered sets, in I. Rival [181], 125–170.

[24] Bonne, W. W., Certain simple unsolvable problems in group theory, I–VI Nederlakad. Wentenschappen, Proc. Ser. A, vol. 57 (1954), 231–237, 492–497, vol. 58 (1955), 252–256, 571–577, vol. 60 (1957), 22–27, 227–232.

[25] Calhoune, W.C., *The Lattice of Ideas of Recursively Enumerable Degrees*, Dh.D.Diss, Berkeley, 1992.

[26] Calhoune, W.C., Incomparable prime ideals of recursively enumerable degrees, (to appear).

[27] Cenzer, D. and J. B. Remmel, Polynomial time vs recursive models, Ann. Pure and Applied Logic, Vol. 54 (1991), 17-58.

[28] Chang, C. C. and H. J. Kiesler, *Model Theory*, North Holland, Amsterdam (1973).

[29] Chen, K. H., Recursive well founded linear orderings, Ann. Math. Logic, vol. 13 (1978), 117–147.

[30] Cholak, P., S. Goncharov, B. Khoussainov, and R. Shore, On recursively categorical models, in preparation.

[31] Chisholm J., and M. Moses, Undecidable linear orderings that are n-recursive for each n, in preparation.

[32] Craven, T., The boolean space of orderings of a field, Trans. Amer. Math. Soc. (1975), 289–371.

[33] Crossley, J.N., *Constructive Order Types*, North-Holland, Amsterdam, (1969).

[34] Crossley, J.N., Recursive equivalence, (survey paper) Bull. London Math. Soc. Vol. 2, (1970), 129-151.

[35] Crossley, J. N., *Aspects of Effective Algebra*, (Ed) Upside Down A Book Co., Varra Glen, Vic. Australia (1981).

[36] Crossley, J.N., (ed) Reminiscenses of Logicians II, In Crossley [35], 1-25.

[37] Crossley, J.N. and Nerode, A., *Combinatorial Functions*, Springer-Verlag, New York, (1974).

[38] Crossley, J.N., Manaster, A. and Moses, M., Recursive stability and recursive categoricity, Ann. Pure and Appl. Logic, Vol 31, (1986), 101-204.

[39] Crossley, J.N. and Parikh, R.J., On isomorphisms of recursive well orderings, (abstract), J. Symb. Logic, Vol. 28, (1963).

[40] Davis, M. and Weyuker, E. *Computability, Complexity and Languages*, Academic Press, New York (1983).

[41] Dehornoy, P., Turing complexity of the ordinals, Information Processing Letters, Vol. 23 (1986), 167-170.

[42] Dehn, M, Über unendlische diskontinuierliche gruppen, Annals of Math., Vol. 71 (1911), 116-144.

[43] Dekker, J. C. E. and J. Myhill, *Recursive Equivalence Types*, University of California Publ., vol. 3 (1960), 67–214.

[44] Dilworth, R. P., A decomposition theorem for partially ordered sets, Ann. of Math., vol. 51.

[45] Downey, R. G., *A Problem of Rosenstein*, unpublished manuscript, (1985), 161–166.

[46] Downey, R., Every recursive boolean algebra is isomorphic to one with incomplete atoms, Ann. Pure and Appl. Logic, Vol. 60 (1990), 193-206.

[47] Downey, R., Recursion theory and linear orderings, *Handbook of Computable Algebra,* to appear. (Available as Victoria University Research Report 91-71, June 1991.)

[48] Downey, R. and C. Jockusch, T-degrees, jump classes and strong reducibilities, Trans. Amer. Math. Soc., vol. 301 (1987), 103-137.

[49] Downey, R. and C. Jockusch, Every low boolean algebra is isomorphic to a recursive one, Proc. Amer. Math. Soc., Vol. 122, (1994), 871-880.

[50] Downey, R. G. and J F. Knight, Orderings with α–th jump degree $\mathbf{0}^{(\alpha)}$, Proc. Amer. Math. Soc., vol. 14 (1992), 545–552.

[51] Downey, R.G. and Kurtz, S.A., Recursion theory and ordered groups, Ann. Pure and Appl. Logic, Vol. 32 (1986), 137-451.

[52] Downey, R.G. and Moses, M.F. On choice sets and sstrongly nontrivial self-embeddings of recursive linear orderings, Z. Math. Logik Grunall. Math, Vol. 35 (1989), 237-246.

[53] Downey, R.G. and Moses, M.F. Recursive linear orderings with incomplete successivities, Trans. Amer. Math. Soc. Vol. 326, 653-668.

[54] Dushnik, B. and Miller, E.W., Concerning similarity transformations of linearly ordered sets, Bull. Amer. Math. Soc. Vol. 46 (1940, 322-326.

[55] Dushnik, B. and Miller, E.W., Partially ordered sets, Amer. J. Math. Vol. 63, (1941), 600-610.

[56] Dzgoev, V.D. and Goncharov, S.S., Autostability of models, Algebra and Logic, Vol. 19, (1980), 13-27.

[57] Eisenberg, E., *Effective Isomorphisms of Groups and Other Structures*, Thesis, Cornell University, (1974).

[58] Eisenberg, E. and Remmel, J.B., Effective isomorphisms of algebraic structures, in Metakides (1982, 85-122.

[59] Ershov, Y., S. Goncharov, A. Nerode, and J. Remmel, (eds) *Handbook of Computable Algebra,* to appear.

[60] Ellentuck, E., The first order properties of Dedekind finite integers, Find. Math. Vol. 63, (1968), 7-25.

[61] Feldman, A., *Recursion theory in a lower semilattice*, J. Symb. Logic, Vol. 57, (1992), 892-911.

[62] Fellner, S., Recursive and finite axiomatizability of linear orderings, Thesis, Rutgers University (1976).

[63] Feiner, L. J., *Orderings and Boolean Algebras not isomoprhic to recursive ones*, Thesis, MIT (1967).

[64] Feiner, L. J., Hierarchies of boolean algebras, J. Symb. Logic, vol. 35 (1970), 365–373.

[65] Feiner, L. J., The strong homogeneity conjecture, J. Symb. Logic, vol. 35 (1970), 373–377.

[66] Feiner, L. J., Degrees of non–recursive presentability, Proc. Amer. Math. Soc., vol. 38 (1973), 621–624.

[67] Friedberg, R. M., Two recursively enumerable sets of integers with incomparable degrees of unsolvability, Proc. Nat. Acad. Sci. U.S.A., vol. 43 (1957), 236–238.

[68] Friedberg, R.M., Three theorems on recursive enumeration, J. Symb. Logic, Vol. 23, (1958), 209-216.

[69] Friedberg, R.M. and Rogers, H. Reducibility and Completeness for sets of integers, Z. Math. Logik Grundl. Math. Vol. 5, (1959), 117-125.

[70] Friedman, H. and J. Hirst, Weak comparibility of well–orderings and reverse mathematics, Ann. Pure and Appl. Logic, vol 47 (1990), 11–30.

[71] Friedman, H., S. Simpson and R. Smith, Countable algebra and set existence axioms, Ann. Pure and Appl. Logic, vol. 25 (1983), 141–181.

[72] Fuchs, L., *Infinite Abelian Groups, Vol. II*, Academic Press, 1973.

[73] Gasarch, W. I., A hierarchy of functions with applications to recursive graph theory, University of Maryland Technical Report TR 1651.

[74] Gasarch, W. I. and M. Lockwood, The existence of matchings for recursive and highly recursive bipartite graphs, University of Maryland Technical Report TR 2029.

[75] Gasarch, W. I., D. Keuker and D. Mount, Highly recursive rooted graphs, Congressus Numerantium, vol. 69 (1989), 97-102. (also in the 19th Southeastern Conference on Combinatorics, Baton Rouge, 1988)

[76] Garsach, W.I., Keuker, D. and Mount, D., Recursive categoricity of highly recursive graphs, (to appear).

[77] Goncharov, S. S., Constructivizability of superatomic boolean algebras, Algebra and Logic, Vol. 12, (1972), 17-22.

[78] Goncharov, S. S., Some properties of the constructivizaton of Boolean algebra, Sibirski Math. Zh., vol. 16 (1975), 264–278.

[79] Goncharov, S. S., On the number of nonautoequivalent constructivizations, Algebra snd Logic, Vol. 16 (1977), 169-185.

[80] Goncharov, S. S., Autostability of models and abelian groups, Algebra and Logic, Vol. 19 91980), 13-27.

[81] Goncharov, S. S. and A. Nurtazin, Constructive models of complete solvable theoreis, Algebra and Logic, vol. 12 (1973), 67–77.

[82] Grigoreiff, S., Every recursive linear ordering has a copy in DTIME–SPACE $(n, \log(n))$, J. Symb. Logic, vol. 55 (1990), 260–276.

[83] Harizanov, V., Some effects of Ash–Nerode and other decidability conditions on degree spectra, Ann. Pure Appl. Logic. vol. 54 (1991), 51-65.

[84] Harizanov, V., Uncountable degree spectra, Ann. Pure Appl. Logic. vol. 54 (1991), 255-263.

[85] Harizanov, V., Turing degree of the non-zero member in a two element degree spectrum , Ann. Pure and Appl. Logic, vol. 60 (1993), 1-30.

[86] Harrington, L., Building nonstandard models of Peano Arithmetic, notes (1979).

[87] Hay, L., A. Manaster and J. Rosenstein, Concerning partial recursive similarity transformations of linearly ordered sets, Pacific J. Math., vol. 71 (1977), 57–70.

[88] Higman, G., Subgroups of finitely presented groups, Proc. Royal Soc. London, Vol. 262 (1961), 455-475.

[89] Higman, G. and E. Scott, *Existentially Closed Groups,* Oxford University Press, 1988.

[90] Hird, G., *Recursive Properties of Relations on Models,* Ph.D. Thesis, Monash University (1983).

[91] Hirst, J., Marriage theorems and reverse mathematics, in *Contemporary Mathematics,* Vol. 106, A.M.S. Publ. Providence R.I. (1990), 181-196.

[92] Hölder, O., Die axiome der quantität urd die lehre vom mass – Ber. Verhy. Sächs. Ges. Wiss. Leipsig, Math. Phys. Kl. vol. 53, (1901) 1-64.

[93] Hiraguchi, T., On the dimension of orders, Sci. Report. on Kaazawa Univ. Vol. 4, (1955), 1-20.

[94] Hopkins. L., *Some Problems Involving Combinatorial Structures Determined by Intersectins of Intervals and Arcs,* Thesis, Univ. of South Carolina, (1981).

[95] Jockusck, C.G., *Reducibilities in Recursive Function Theory,* Thesis, MIT, (1966).

[96] Jockusch, C.G., Semirecursive sets and positive reducibility, Trans. Amer. Math. Soc. Vol. 131, (1968), 420-438.

[97] Jockusch, C. J., Relationships between reducibilities, Trans. American Math. Soc., vol. 142 (1969), 229-237.

[98] Jockusch, C. G., Ramsey's theorem and recursion theory, J. Symb. Logic, vol. 37 (1972), 268–279.

[99] Jockusch, C. G. and R. I. Soare, Degrees of members of $\prod_1^0$ classes, Pacific J. Math. Vol. 40, (1972), 605-616.

[100] Jockusch, C. G. and R. I. Soare, $\prod_1^0$ classes and degrees of theories, Trans. Amer. Math. Soc., Vol. 173, (1972), 33-55.

[101] Jockusch, C. G. and R. I. Soare, Degrees of orderings not isomorphic to recursive linear orderings, Ann. Pure and Appl. Logic., Vol. 52 (1991), 39-64.

[102] Jockusch, C. and R. Soare, Boolean algebras, Stone spaces, and the iterated Turing jump, Journal of Symbolic Logic, Vol. 59 (1994), 1121-1138.

[103] Kelly, D. and Trotter, W.T., Dimension theory for ordered sets, in I. Rival, (1981), 171-211.

[104] Ketonen, J., The structure of countable boolean algebras, Ann. of Math., Vol. 108 (1978), 41-79.

[105] Khoussainov, B and A. Nerode, On automata representable structures, in preparation.

[106] Khoussainov B. and A. Nerode, *Automatic Model Theory*, monograph in preparation.

[107] Khisamiev, N., Connections between constructivizability and nonconstructavizability for different classes of abelian groups, Algebra and Logic, Vol. 23 (1984), 220-233.

[108] Khisamiev, N., Hierarchies of torsion free abelian groups, Algebra and Logic, vol. 25 (1986), 128-142.

[109] Khisamiev, N., and Z. Khisamiev, Nonconstructavizability of the reduced part of a strongly constructive torsion-free abelian group, Algebra and Logic, Vol. 24 (1985), 69-76.

[110] Kierstead, H., An effective version of Dilworth's theorem, Trans. Amer. Math. Soc., vol. 268 (1981), 63–77.

[111] Kierstead, H., Recursive colorings of highly recursive graphs, Can. J. Math., vol. 33 (1981), 1279–1290.

[112] Kierstead, H., An effective version of Hall's theorem, Proc. Amer. Math. Soc., vol. 88 (1983), 124–128.

[113] Kierstead, H., Recursive ordered sets, in I. Rival (ed) *Combinatorics and Ordered Sets*, Contemporary Math., vol 57, Amer. Math. Soc. (1986).

[114] Kierstead, H., On Π_1–automorphisms of recursive linear orderings, J. Symb. Logic, vol. 52 (1987), 681–688.

[115] Kierstead, H. and Trotter, W., An extrema problem in recursive combinatorics, Congress Numerantium, vol. 33 (1981), 143–153.

[116] Kierstead, H., G. McNulty and W. Trotter, Recursive dimension for partially ordered sets, Order, vol. 1((1984), 67–82.

[117] Knight, J. F., Effective constructions of models, in *Logic Colloquium* (ed. Paris, Wilkie and Wilmers) North–Holland, Amsterdam (1986).

[118] Knight, J. F., Degrees coded into jumps of orderings, J. Symb. Logic, vol. 51 (1986), 1034–1042.

[119] Knight, J.F., Degrees of models with prescribed Scott set, in *Classification Theory: Proc. Chicago 1985* (ed. J. Baldwin), Springer-Verlag, (1988), 182-191.

[120] Knight, J. F., A metatheorem for finitely many workers, J. Symbolic Logic, vol. 55 (1990), 787–804.

[121] Knight, J. F., Constructions for transfinitely many workers, Ann. Pure and Appl. Logic, vol. 48 (1990), 237–259.

[122] Kopytov, A., and V. Kokorin, *Fully Ordered Groups,* John Wiley and Sons, 1974.

[123] Kriesel, G., J. Shoenfield and H. Wang, Number theoretic concepts and recursive well orderings, Arch. Math. Logik Grund., vol. 5 (1960), 42-64.

[124] Kruskal, J., Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture, Trans. Amer. Math. Soc., Vol. 95, (1960) 210-220

[125] Lachlan, A.H. Lower bounds for pairs of recursively enumerable degrees, Proc. Lond. Math. Soc., Vol. 16, (1966), 537-569.

[126] Langford, C. H., Theorems on deducibility, Ann. of Math., vol. 28 (1927), 459–471.

[127] LaRoche, P., Recursively presentable Boolean algebras, Not. Amer. Math. Soc., vol. 24 (1977), A–552.

[128] Lavev, R., On Fraisse's order type conjecture, Ann. of Math., Vol. 93, (1971), 89-111.

[129] Lempp, S. and M. Lerman, Priority arguments in interated trees of strategy, in *Recursion Theory Week* (Proceedings Oberwolfach (1989)), Springer–Verlag, New York (1990).

[130] Lerman, M., On recursive linear orderings, in Lerman, Schmerl and Soare [1981], 132–142.

[131] Lerman, M., *Degrees of Unsolvability*, Springer–Verlag, New York (1983).

[132] Lerman, M. and J. Rosenstein, Recursive linear orderings, in Metakides [1982], 123–136.

[133] Lerman, M. and J. Schmerl, Theories with recursive models, J. Symb. Logic, vol. 44 (1979), 59–76.

[134] Lerman, M., J. Scherml and R. Soare, *Logic Year 1979–80*, Lecture Notes in Mathematics, vol. 850, Springer–Verlag, New York (1981).

[135] Lin, C., Recursion theory on countable abelian groups, Ph. D. Thesis, Cornell University, 1977.

[136] Lin, C., The effective content of Ulm's theorem, in Crossley [35], 147-160.

[137] Macintyre, A., Transfinite extensions of Friedberg's completeness criterion, J. Symb. Logic, vol. 42 (1977), 1–10.

[138] Manaster, A. and J. B. Remmel, Some recursion theoretic aspects of dense two dimensional partial orderings, in Crossley [1981], 161–188.

[139] Manaster, A. and J. B. Remmel, Partial orderings of fixed finite dimension: model comparisons and density, J. Symb. Logic, vol. 46 (1981), 789–802.

[140] Manaster, A. and J. Rosenstein, Effective matchmaking (recursion theoretical aspects of a theorem of P. Hall), Proc. London Math. Soc., vol 25 (1971), 615–654.

[141] Manaster, A. and J. Rosenstein, Two dimensional partial orderings, recursive model theory, J. Symb. Logic, vol. 45 (1980), 121–132.

[142] Marker, D., Degrees of models of true arithmetic, in *Proceedings Herbrand Symposio,* (J. Stern, ed.), North-Holland, Amsterdam, (1982), 233-242.

[143] Martin, D.A., Classes of recursively enumerable sets, Trans. Amer. Math. Soc., Vol. 130, (1968), 1-27.

[144] McCarty, C., Realizability and recursive set theory, Ann. Pure and Appl. Logic, Vol. 32, (1986), 153-183.

[145] McNulty, G., Infinite ordered sets, a recursive perspective, in I. Rival (ed) *Proc. of Symp. on Ordered Sets*, D. Reidel, Dortrecht (1982).

[146] Mead, J., Recursive prime models for boolean algebras, Colloq. Math., Vol. 41 (1979), 25-33.

[147] Metakides, G., *Patras Logic Symposium*, North–Holland, Amsterdam (1981).

[148] Metakides, G. and A. Nerode, Recursion theory and algebra, in *Algebra and Logic* (ed. J. N. Crossley), Lecture notes in Math., vol. 450, New York (1975), 209–219.

[149] Metakides, G. and A. Nerode, Recursively enumerable vector spaces, Ann. Math. Logic, Vol. 11 (1977), 141-171.

[150] Metakides, G. and A. Nerode, Effective content of field theory, Ann. Math. Logic, vol. 17 (1979), 289–320.

[151] Metakides, G. and J. B. Remmel, Recursion theory on orderings I: a model theoretic setting, J. Symb. Logic, vol. 44 (1979), 383–402.

[152] Minsky, M., Recursive unsolvability of Post's problem of "tag" and other topics in the theory of Turing Machines, Ann. of Math. (2) Vol. 74, (1961), 437-455.

[153] Minsky, L., A dual to Dilworth's decomposition theorem, Amer. Math. Monthly, Vol. 78, (1971), 876-877.

[154] Morozov, A., Countable homogeneous boolean algebras, Algebra and Logic, Vol. 21. (1982), 181-190.

[155] Morozov, A., Strong constructivizability of countable saturated boolean algebras, Algebra and Logic, Vol. 21 (1982), 130-137.

[156] Morozov, A., Groups of computable automorphisms, to appear.

[157] Moses, M. F., *Recursive Properties of Isomorphism Types*, Thesis, Monash University, Melb. Australia (1983).

[158] Moses, M. F., Recursive linear orders with recursive successivities, Ann. Pure and Appl. Logic, vol. 27 (1984), 253–264.

[159] Moses, M. F., Relations intrinsically recursive in linear orderings, Z. Math. Logik Grundlagen. Math., vol. 32 (5), 467–472.

[160] Moses, M. F., Decidable discrete linear orderings, J. Symb. Logic, vol. 53 (1988), 531–539.

[161] Moses, M. F., n-recursive linear orderings without $n+1$-recursive copies, in *Logical Methods,* (Ed. J. Crossley, J. Remmel, R. Shore and M. Sweedler), Birkhäuser (1993), 572-592.

[162] Muchnik, A. A., On the unsolvability of the problem of reducibility in the theory of algorithms, Dokl. Akad. Nauk SSSR, vol. 108 (1956), 194–197.

[163] Nerode, A., Extensions to isols, Ann. of Math., Vol. 73, (1961), 362-403.

[164] A. Nerode and J. Remmel, Complexity-theoretical algebra II, the free boolean algebra, Annals Pure and Appl. Logic, Vol. 41 (1989), 71-99.

[165] Nerode, A. and Remmel, J.B., Polynomial equivalence types, in *Contemporary Mathematics,* Vol. 106, Amer. Math. Soc., Providence, R.I., (1990).

[166] Neumann, B., On ordered groups, Amer. J. Math., vol. 71 (1949), 1–18.

[167] Neumann, B., An embedding theorem for ordered groups, J. Lond. Math. Soc., vol. 35 (1960), 503–512.

[168] Nurtazin, A., Strong and weak constructivizations and computable families, Algebra and Logic, Vol. 13 (1974), 177-184.

[169] Oates, S., *Jump Degrees of Groups*, Ph. D. Thesis, University of Notre Dame, 1989.

[170] Odifreddi, P., *Classical Recursion Theory, Vol. I*, North–Holland (1989).

[171] Paris, J. and Harrington, L., A mathematical incompleteness in Peano arithmetic, in *Handbook of Mathematical Logic* (J. Barwise ed.), North-Holland, (1977), 1133-1142.

[172] Peretyat'kin, M., Every recursively enumerable extension of a theory of linear orderings has a constructive model, Algebra i Logik, vol. 12 (1973), 211–219. '

[173] Pinus, A., Effective linear orderings, Siberian Math. J., vol. 16 (1975), 956–962.

[174] Post, E. L., Recursively enumerable sets of integers and their decision problems, Bull. Amer. Math. Soc., vol. 50 (1944), 284–316.

[175] Rabin, M. O., Computable algebra, general theory and theory of computable fields, Trans. Amer. Math. Soc., vol. 95, (1960), 341–360.

[176] Raw, M., Ph. D. Thesis, University of Wisconsin, Madison, 1995.

[177] Remmel, J. B., Recursively enumerable boolean algebras, Ann. Math. Logic, vol. 14 (1978), 75–107.

[178] Remmel, J. B., Recursion theory on algebraic structures with an independent set, Ann. Math. Logic, vol. 18 (1980), 153–191.

[179] Remmel, J. B., Recursion theory on orderings II, J. Symb. Logic, vol. 45 (1980), 317–333.

[180] Remmel, J. B., Recursive boolean algebras with recursive atoms, J. Symb. Logic, vol. 46 (1981), 595–615.

[181] Remmel, J. B., Recursive isomorphism types of recursive boolean algebras, J. Symb. Logic, vol. 46 (1981), 572–594.

[182] Remmel, J. B., Recursively categorical linear orderings, Proc. Amer. Math. Soc., vol. 83 (1981), 379–386.

[183] Remmel, J. B., On the effectiveness of the Shröder–Bernstein theorem, Proc. Amer. Math. Soc. (1983).

[184] Remmel, J. B., Graph coloring and recursively bounded Π_1^0 classes, Ann. Pure and Appl. Logic, vol. 32 (1986), 185–194.

[185] Remmel, J. B., Recursively rigid boolean algebras, Ann. Pure and Appl. Logic, vol. 36 (1987), 39–52.

[186] Remmel, J. B., Recursive Boolean algebras, in *Handbook of Boolean Algebras, Vol 3* (ed. D. Monk) North Holland (1990), 1097-1166.

[187] Rice, H., Recursive and recursively enumerable orders, Trans. Amer. Math. Soc., vol. 83 (1956), 277–300.

[188] Richter, L, *Degrees of Structures*, Ph. D. Thesis, University of Illinois at Urbana-Champaign, 1979.

[189] Richter, L. J., Degrees of structures, J. Symb. Logic, vol. 46 (1981), 723–731.

[190] Richter, L. J., On automorphisms of the degrees preserving jumps, Israel J. Math., Vol. 32 (1979), 27-31.

[191] Rival, I., *Ordered Sets* (ed.) Nato Advanced Study Inst. Series 83, D. Reidel Co., Dordrecht (1981).

[192] Rival, I., *Graphs and Order* (ed.) Nato Advanced Study Inst. Series, vol. 147, D. Reidel Co., Dordrecht (1984).

[193] Rival, I., *Combinatorics and Ordered Sets* (ed.), Contemporary Math., vol. 57, Amer. Math. Soc. (1986).

[194] Rogers, H. J., *Theory of Recursive Functions and Effective Computability*, McGraw–Hill, New York (1967).

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

[196] Rosenstein, J., Recursive lienar orderings, in *Orders: Descriptions and Roles* (M. Pouer and P. Richards, eds.), Ann. Discrete Math., vol. 23, North–Holland (1982), 465–476.

[197] Roy, D. K., R.e. presented linear orders, J. Symb. Logic, vol. 48 (1983), 369–376.

[198] Roy, D. C., Linear orderings of nonrecursive presentability, Z. Math. Logik. Grundl. Math., vol. 31 (1985), 493–501.

[199] Roy, D. K. and R. Watnik, Finite condensations of recursive linear orderings, Studia Logika, vol. 47 (1988), 311–317.

[200] Sacks, G. E., The recursively enumerable degrees are dense, Ann. of Math., vol. 80 (1964), 300–312.

[201] Salomaa, A., *Computation and Automata*, Cambridge Univ. Press (1985).

[202] Schwartz, S., Quotient lattices, index sets and recursive linear orderings, Thesis, Chicago University (1982).

[203] Schwartz, S., Recursive automorphisms of recursive linear orderings, Ann. Pure and Appl. Logic, vol. 26 (1984), 69–73.

[204] Simpson, S., *Logic and Combinatorics*, (ed.) Contemporary Math., vol. 65, Amer. Math. Soc., Rhode Island (1986).

[205] Simpson, S. Unprovable theorems and fast-growing functions, in Simpson [204], 359-394.

[206] Simpson, S. *Subsystems of Second Order Arithmetic,* in preparation.

[207] Smith, R., Two theorems on autostability in p-groups, in Lerman, Schmerl, Soare [134], (1981) 302-311 .

[208] Soare, R. I., Recursion theory and Dedekind cuts, Trans. Amer. Math. Soc., vol. 140 (1969), 271–294.

[209] Soare, R. I., Constructive–order types on cuts, J. Symb. Logic, vol. 34 (1969), 285–289.

[210] Soare, R.I., Automorphisms of the lattice of recursively enumerable sets, part I: maximal sets, Ann. of Math. Vol. 100, (1974), 80-120.

[211] Soare, R.I., The infinite injury priority method, J. Symb. Logic, Vol. 41, (1976), 513-550.

[212] Soare, R.I., Computational complexity speedable and levelable sets, J. Symb. Logic, Vol. 42, (1977), 545-563.

[213] Soare, R. I., *Recursively Enumerable Sets and Degrees*, Springer–Verlag, New York (1987).

[214] Szpilrajin, S., Sur les tension de l'orde partial, Fund. Math., vol. 45 (1958), 213–216.

[215] Schmerl, J., Recursive colorings of graphs, Can. J. Math., vol 32 (1980), 821–883.

[216] Schmerl, J., The effective version of Brooks' theorem, Can. J. Math., vol. 34 (1982), 1036–1046.

[217] Schmerl, J., Recursion theoretical aspects of graphs and orders, in I. Rival [1984], 467–486.

[218] Szmielew, W., Elementary properties of abelian groups, Fund. Math., vol. 41 (1955), 203–271.

[219] Thurber, J., *Degrees of Boolean Algebras,* Ph. D. Thesis, University of Notre Dame, 1994.

[220] Trotter, W. and T. Moore, A combinatorial problem involving graphs and matrices, Discrete Math., vol. 39 (1982), 87–101.

[221] Trotter, W. and J. I. Moore, Characterization problems for graphs: partially ordered sets, lattices and families of sets, Discrete Mathematics, vol. 16 (1976), 361–381.

[222] Tverberg, H., On Schmerl's effective version of Brook's theorem, J. Comb. Theory (Ser B), vol. 37 (1984), 27–30.

[223] Watnik, R., *Recursive and constructive linear orderings*, Thesis, Rutgers University (1980).

[224] Watnik, R., Constructive and recursive scattered order types, in Lerman, Schmerl and Soare [134], 312–326.

[225] Watnik, R., A generalization of Tennenbaum's theorem on effectively finite recursive linear orderings, J. Symb. Logic, vol. 49 (1984).

[226] Yates, C.E.M., A minimal pair of recursively enumerable degrees, J. Symb. Logic, (1966), 159-168.

[227] Young, D. R., Linear orderings under 1–1 reducibility, J. Symb. Logic, vol. 31 (1966), 70–85.