Automorphisms and Linearisations of Computable Orderings

Kyung Il Lee

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy

The University of Leeds
School of Mathematics

September 2011

Acknowledgements

The gratitude for the compassionate help and encouragement during the PhD study is thanks to my supervisor Professor S. Barry Cooper. I would also like to express my hearty thanks to Dr. Charles M. Harris and Dr. Anthony W. Morphett for their useful advise and intimate friendship. And many thanks to all the people whom I met in lovely Leeds. Professor Iraj Kalantari (at Western Illinois University) and Professor Inkyo Chung (at Korea University) introduced the beautiful subjects: computability theory and philosophical logic to me before the PhD study. Many thanks for their inspiration and guidance. Special thanks to my parents, my wife Hyun Jin and my daughter Chae-Un for their endless love, support and encouragement.

Abstract

In this thesis, we study computable content of existing classical theorems on linearisations of partial orderings and automorphisms of linear orderings, and provide computational refinements in terms of the Ershov hierarchy. In **Chapter 2**, we examine questions as to the constructiveness of linearisations obtained in terms of the Ershov hierarchy, while respecting particular constraints. The main result here entails a proof that every computably well-founded computable partial ordering has a computably well-founded ω-c.e. linear extension. In **Chapter 3**, we examine questions as to how less constructive rigidities of certain order types break down within the context of the Ershov hierarchy, and introduce uniform Δ^0_2 classes as likely candidates in the case of order types $\mathbf{2} \cdot \boldsymbol{\eta}$ and $\boldsymbol{\omega} + \boldsymbol{\zeta}$.

Contents

To my parents,

my wife Hyun Jin

and my daughter Chae-Un

Chapter 1

Introduction

The first two sections in this chapter consist of a brief background survey of computability theory and computable orderings. For more extensive sources, we refer to Cooper (2004 [7]), Odifreddi (1989 [45], 1999 [46]), Soare (1987 [60], 2008 [63], 2009 [64]) for computability theory, and Downey (1998 [10]), Rosenstein (1982 [51]) for computable orderings. But no knowledge of these two topics will be assumed. In the third section we give an example introducing the tree of strategies method for structuring a priority argument. And in the last section we give an overview of some basic definitions and results concerning ordinal notations and the Ershov hierarchy.

1.1 Computability Theory

The initial but still notable achievement is, amongst others in computability theory, to capture the intuitive notion of "(effectively) computable function". The notion was formalised through the development of terms such as

"λ-definable" (Church 1931), "general recursive"[1](Gödel 1934, [22]), "recursive"[2](Church and Kleene 1935), and "(Turing) computable" (Turing 1936, [69]), which are the same heuristic concepts although the labelling and the formalisation are different.[3]In fact, the definitions of "λ-definable", "general recursive" and "recursive" were established through *logical reasoning*[4]in the context of mathematics, whereas that of "(Turing) computable" used Turing's *automatic machine* (*a-machine*, also known as **Turing machine**), which is adequate enough to encompass any mechanical computation.

Note that Turing machines provided the first convincing comprehensive enough definition of a computable function in the opinion of Gödel. (For historical remarks on concepts of computability, see Soare's papers: 1996 [61], 2007 [62], 2009 [65] and 2012 [66].) Notice that the story so far is only for (partial) computable functions which range over numbers. Now the central device in computability theory, Turing's *oracle machine* (*o-machine*, also known as **oracle Turing machine**) was introduced by Alan Turing [70] in 1939. The notion of oracle Turing machines leads us to look at functionals[5]in an effective sense. Stephen Kleene and Leonard Sasso were amongst the first ones to recognise the importance of this notion, generalising it into the natural notion, relative *partial recursive functionals*.

[1]The term "recursive" had referred to "inductive" by the general population before 1930's and to "primitive recursive" by Kurt Gödel in the early of 1930's, and the idea of the term "general recursive" was introduced by Jacques Herbrand (1931).

[2]Alonzo Church and Stephan Kleene changed the meaning of the term "recursive" from *primitive recursive* to *(effectively) computable* after Gödel delivered a lecture on general recursive functions at Princeton in 1934.

[3]Church-Turing Thesis

[4]The definitions may be conceived of as intensional meanings of "inductively defined" through logical reasoning.

[5]A functional is a certain general type of function whose variables range over numbers or functions of numbers, and whose values are numbers.

Definition 1.1 (Kleene 1952 [30], 1969 [32], Sasso 1971 [54]; see Odifreddi [45] p. 178).[6] The functional $F(\alpha_1, \ldots, \alpha_n, \overrightarrow{x})$ is a ***partial recursive functional*** if it can be obtained from partial functions[7] $\alpha_1, \ldots, \alpha_n$ (*oracles*) and the *initial functions* by *composition*, *primitive recursion* and *unrestricted* μ*-recursion.*

Since a partial recursive functional of variable α_1

$$\lambda\alpha_1.F(\alpha_1, \ldots, \alpha_n, \overrightarrow{x})$$

is uniformly (partial) recursive — having a master way to compute — in parameters $\alpha_2, \ldots, \alpha_n$ and $\overrightarrow{x}$, there is an effective listing of functionals $\{\lambda\alpha_1.F_e\}_{e\in\omega}$ from functions to functions, giving the most important relation in computability theory "computable from", written $\beta \leq_T \alpha_1$ for $F_i(\alpha_1) \simeq \beta$ [8], some $i \in \omega$, and called "Turing reducible to". The important point is that this notion provides the basics for the *Turing degree structure*; namely, Cantor space measured by its degree under *Turing (decision problem) reducibility.*

In fact, choosing a set of natural numbers[9] (or equivalently, a binary real), $A \in 2^\omega$ say, which is identified with the corresponding characteristic (total) function, as oracle and obtaining the output Φ^A — giving the notion of *Turing functional* (Turing 1939, [70]) — applying the *signum function*[10] at the last computation, we can capture *information content* of the binary real A. In this case, the (preorder) relation $\leq_T$ defines an equivalence relation $\equiv_T$, which

[6] A partial recursive functional was got by adding oracles $\alpha_1, \ldots, \alpha_n$ to the initial functions. In other words, it was got by a uniformisation of a function $\lambda\overrightarrow{x}.F(\alpha_1, \ldots, \alpha_n, \overrightarrow{x})$, which indeed becomes convincing computation in invoking an oracle Turing programs. One remark is that Kleene (1978, [33]) developed the reversal approach to define partial recursive functions from partial recursive functionals by using *the First Recursion Theorem* together with *composition* and *case definition*. For details, see Odifreddi (1987 [45], pp. 174–184).

[7] The partial functions $\alpha_1, \ldots, \alpha_n$ range over numbers.

[8] An extended equality relation $\simeq$ indicates either both are undefined or defined with the same value.

[9] In computability theory, a *set* simply refers to a set of natural numbers.

[10] The signum function is defined by sg$=\lambda x.1$ if $x \neq 0$; sg$=\lambda x.0$ otherwise.

partitions the class of characteristic sets (i.e. sets identified with characteristic functions)[11]into the equivalence classes, and induces a partial ordering $\leq$ on those classes (Post 1944, [48]). Such classes are said to be ***Turing degrees*** or ***degrees of*** (***algorithmic***) ***unsolvability*** and the ***Turing degree structure*** $\langle \boldsymbol{\mathcal{D}}, \leq \rangle$ denotes the structure of Turing degrees, with the partial ordering $\leq$ induced on them (Kleene and Post 1954, [34]).

To obtain *intrinsic* properties of $\langle \boldsymbol{\mathcal{D}}, \leq \rangle$[12]and to get definability of the relations on it, we measure *computational complexity* of reals in terms of *(total)*[13]*oracle strength* relative to the *computable constructions*. In virtually all such computable constructions concerning $\langle \boldsymbol{\mathcal{D}}, \leq \rangle$, we observe *acceptable description systems* (or *universal computers*), which can be encoded by natural numbers in an effective way, such as *Turing programs* or *oracle Turing programs*, which we will adopt in this thesis. In other words, *descriptions* such as a list $\{W_e\}_{e \in \omega}$ of the (standard) computably enumerable (c.e.) sets — W_e being the set of inputs on which the e-th Turing program halts — in a description system help a priori to get *descriptive complexity* of such constructions. The *priority method* is a common part of such constructions, which was first required in constructing Turing incomparable (so incomplete) c.e. sets (Friedberg 1957, [19] and Mučnik 1956, [43]).[14]

The *tree of strategies method* for priority arguments is one of the unifying frameworks for approximating classical[15]*proof* of such constructions, which was introduced by Alistair Lachlan (1975 [36]) and Leo Harrington (1982 [25]).

[11]It can be easily generalised with the class of all total functions in place of that of sets.

[12]"The level of the method needed to prove that a given sentence is true is closely related to the logical complexity of the sentence." (*A Framework for Priority Arguments* by Manuel Lerman, page 2, 2007, http://www.math.uconn.edu/∼lerman/GFposet.pdf)

[13]In the case of enumeration degrees, computational complexity indicates *partial* oracle strength.

[14]Along with the constructions, usual mathematical practices are required, which we call *verifications*.

[15]For example, not intuitionistic.

The best introductory and expository paper on the tree method is by Robert Soare (1985 [59]). On the other hand, there is a very powerful framework for approximating *truth*: *forcing*. In contrast to forcing, the following example stresses the connotational (intrinsical) importance of the priority method.

Theorem 1.2 (Jockusch and Posner; [60] Exercise VI.3.8)**.** *Every* 1*-generic set — i.e. which forces a c.e. set — is hyperimmune.*

One can merely suspect that 1-generic sets can give any stronger information content than the hyperimmune sets have, which is captured by the subsumed notions of "hyperhyperimmuneness", strong "hyperhyperimmuneness" and "cohesiveness", while a construction of existence of a *maximal* set (complement of a cohesive set) was carried out using the priority method by Richard Friedberg [20] in 1958. Accordingly, the essential distinction between the following notions will provide a key tension throughout the next section on computable orderings:

classical	:	effective
structural	:	computability theoretic
extensional	:	intensional
extrinsic (denotational)	:	intrinsic (connotational)

Notation. We will follow standard notation for computability theory as in Cooper (2004 [7]), Soare (1987 [60], 2008 [63], 2009 [64]). The set of natural numbers is denoted by $\omega = \{0, 1, 2, \ldots\}$. Let Λ be a countable set of any objects. We say that f is a (partial) function *on* ω if $f \in \Lambda^\omega$. Strings in $\Lambda^{<\omega}$ are often denoted by lower-case Greek letters σ, τ, etc. $f(x)\downarrow$, $\sigma(x)\downarrow$ ($f(x)\uparrow$, $\sigma(x)\uparrow$) mean that f, σ is defined (undefined) on x. We say τ *extends* σ (or σ *is an initial segment of* τ) if $\sigma \subseteq \tau$, namely for all $x < |\sigma|$ if $\sigma(x)\downarrow$ then $\tau(x)\downarrow = \sigma(x)$, and we say τ *properly extends* σ if $\sigma \subset \tau$, namely $\sigma \subseteq \tau$ except that for $x = |\sigma|$ it is not the case that $\sigma(x) = \tau(x)$. $\sigma\widehat{\ }\tau$ denotes the concatenation of σ followed by τ. The domain and the range of f are denoted by $\mathrm{dom}(f)$ and $\mathrm{range}(f)$ respectively. We use $\langle \cdot, \cdot \rangle : \omega \times \omega \to \omega$ to denote the standard computable pairing function with computable inverse functions $(\cdot)_0$ and $(\cdot)_1$ — i.e. satisfying the equality $\langle (n)_0, (n)_1 \rangle = n$ for all $n \in \omega$. The complexity of a (partial) function f is that of its graph $G(f) = \{\langle x, f(x) \rangle : x \in \mathrm{dom}(f)\}$. We use $A, B, C, \ldots, X, Y, Z, \ldots$ for sets of natural numbers. We sometimes identify a set A with its characteristic function χ_A. Note, however, that c.e. sets are fundamental objects in computability theory in the sense that all graphs of *partial* computable (p.c.) functions exactly are c.e. sets. We denote the restriction to arguments $y < x$ ($y \leq x$) of f by $f \upharpoonright x$ ($f \upharpoonright\upharpoonright x$) and denote the restriction to elements $y < x$ ($y \leq x$) of A by $A \upharpoonright x$ ($A \upharpoonright\upharpoonright x$). $\overline{A}$ denotes the complement of A. We denote the least natural number such that a relation $R(\overrightarrow{n}, m)$ holds by $\mu m[R(\overrightarrow{n}, m)]$. If no least number exists, $\mu m[R(\overrightarrow{n}, m)]$ is undefined.

Throughout this thesis, we work in the context of a standard computable listing of all Turing machines $\{\varphi_e\}_{e \in \omega}$ mapping ω to ω (i.e. a computable list of all (unary) partial computable functions), with associated computable approximation $\{\varphi_{e,s}\}_{e,s \in \omega}$, where $\varphi_{e,s}(n)$ denotes the result — perhaps undefined — after s stage of the computation of $\varphi_e(n)$. We use $\{\varphi_e^{(2)}\}_{e \in \omega}$ to denote the computable listing of all Turing machines mapping $\omega \times \omega$ into $\{0, 1\}$ derived from $\{\varphi_e\}_{e \in \omega}$ via the pairing function $\langle \cdot, \cdot \rangle$ (i.e. $\varphi_e^{(2)}(n, m) = \varphi_i(\langle n, m \rangle)$ for some i.)[16]

[16]This is a simple application of the s-m-n theorem by S. Kleene:

Theorem 1.3. *If $f(x, y)$ is a p.c. function, then there exists a (total) computable g such that $f(x, y) = \varphi_{g(x)}(y)$.*

1.2 Computable Orderings

We define countable mathematical objects which are *encoded* (*axiomatised*) by some form of mechanical device such as a (oracle) Turing machines.

Definition 1.4. A ***numbering*** (or ***coding*** or ***axiomatisation***) of a set A is a function (possibly partial) from ω onto A. A set A is ***numbered*** (or ***encoded*** or ***axiomatised***) if there is a numbering of A. Note that every countable set has a numbering. (From now on, a set means a subset of ω.) A numbering ν of a set is said to be ***computable*** if the set $\{\langle x, y\rangle : y = \nu(x)\}$ is a c.e. set. For example, standard Kleene's c.e. set of *axioms*: $G_e = \{\langle x, y\rangle : \varphi_e(x) = y\}$ gives a computable numbering of the class of partial computable functions.

Note that these computable codings can be relativised; e.g. in Σ^0_2 theories of linear orderings etc. Now we introduce computable structural relations particularly in linear orderings, which have only one order relation, $<_A$ say.

Definition 1.5. A linear ordering $\langle A, <_A\rangle$ is ***computably presented*** (or just ***computable***) if the domain A is computably numbered and the order relation $<_A$ is computable (equivalently, the atomic diagram of $\langle A, <_A\rangle$ is uniformly computable (normally in A if A is computable)).

We touch on three directions within computable model theory in terms of computable orderings. The first is to explore intrinsic features of computable models. (It plays a role to enlighten *Hilbert's programme*[17](David Hilbert, 1921) not to restrain it within any computable context.) The second is to understand the relationship between classical invariants and computable invariants. (It

[17]Hilbert's programme is to establish a formalisation of all existing theories and of consistency proofs.

can be viewed as a extension of *Erlangen programme*[18](Felix Klein, 1872) in a broad sense.) The last is to locate the complexity of models from decidable[19]to n-computable[20], and further to incomputable.

1. To what extent are intrinsic features of computable models (computable linear orderings)[21]investigated?

To understand this question, we begin with a very easy example. Relative to a structural order relation, we will see how complex its domain is as an intrinsic object.

It is easy to see that if a linear ordering $\langle A, <_A \rangle$ is constructed via a computable approximation for a c.e. set (i.e. computable enumeration) and computable approximation for a c.e. linear ordering, $\langle A, <_A \rangle$ will be computably presented since $<_A^s \subseteq <_A^{s+1}$ so that for all $a, b \in A$ we can effectively decide whether $a <_A b$. The converse also holds:

Proposition 1.6. *If a linear ordering $\langle A, <_A \rangle$ is computably presented — and hence $<_A$ is computable — then A is c.e.*

Proof. Fix some computable numbering of A, and let a_i and a_j be numbered elements of A. Since $<_A$ is computable, we can define a characteristic function $\varphi_e^{(2)}$ of $<_A$ by

$$\varphi_e^{(2)}(a_i, a_j) = \begin{cases} 1 & \text{if } (a_i, a_j) \in <_A, \\ 0 & \text{if } (a_i, a_j) \notin <_A. \end{cases}$$

[18]Erlangen programme is to describe geometry (each branch of mathematics) in terms of a space (a set) and a group of transformations acting on that space.

[19]A structure is ***decidable*** if its whole diagram is computable.

[20]A structure is n-***computable*** if we can decide effectively an n quantifier sentences.

[21]We can restrict the class of computable linear orderings to n-computable linear orderings (up to decidable linear orderings).

But by the s-m-n Theorem,

$$\varphi_{f(a_i,e)}(a_j) = \begin{cases} 1 & \text{if } (a_i, a_j) \in <_A, \\ 0 & \text{if } (a_i, a_j) \notin <_A \end{cases} \quad \text{for some computble } f.$$

Similarly, we get the p.c. function $\varphi_{g(a_j,e)}$ for some computable g.

Therefore $\mathrm{dom}(\varphi_{f(a_i,e)}) \cup \mathrm{dom}(\varphi_{g(a_j,e)})$ $(= A)$ is a computable (so c.e.) set.

□

Thus, in order to get computable linear orderings, its domain must be c.e.

The following series of theorems, as more complicated examples, all relate to which intrinsic natures a theory should take in order to have a computable models.

Theorem 1.7 (Peretyat'kin 1973, [47]). *Every c.e.(Σ_1^0) theory of linear ordering has a computable model.*

Theorem 1.8 (Lerman and Schmerl 1979, [38]). *Every Σ_2^0 theory of linear ordering has a computable model.*

Theorem 1.9 (Lerman and Schmerl 1979, [38]). *There is a Δ_3^0 theory of linear ordering without a computable model.*

In the case of computable linear orderings, Σ_2^0 theories are optimal.

2. **How can isomorphism types be presented effectively**?

This question has been studied not only in relation to particular algebraic structures such as r.e. sets, linear orderings, groups, etc. but also for a wide *class* of structures in a general context. Our interest in this thesis is in self-embeddings or automorphisms of particular computable linear orderings. That is to say, complexity of the graph of self-embeddings is considered up to classical order types. (See **Chapter 3**.)

3. How complex are models (linear orderings)?

There are various areas in which to pursue this question. One such example is to look at complexity of linear orderings which are classically embedded into other computable linear orderings:

Theorem 1.10 (Watnick 1984, [71])**.** *An order type ρ is Π_2 presentable if and only if $\boldsymbol{\zeta} \cdot \rho$ is computably presentable, where $\boldsymbol{\zeta}$ is the order type of integers.*

Our interests in this thesis is another, namely, in complexity of linear orderings which *linearise* a particular computable partial ordering. (See **Chapter 2**.)

Remark 1.11. The conjunction of the questions 2 and 3 can be rephrased as: “How effective is a classical theorem about linear orderings?” (Note that this is very similar to the way in which one asks in *reverse mathematics*: “Which set existence axioms are needed to prove the theorems about linear orderings?”) In fact, in a wider sense, the second and the third directions interplay each other because of, in principal, their commitment to intrinsic computing process, and especially because of the connection between two notions: *computable categoricity*[22]and *intrinsical computability*[23].Two examples are:

[22]Complexity of the graph of isomorphisms of computable linear orderings is considered up to classical order types.

[23]The notion of *intrinsically computable relations* in computable models (up to classical order types) is due to Christopher Ash and Anil Nerode (1981 [3]) and can be relativised, giving *degree spectra of relations* in computable linear orderings, so that a relation R is intrinsically computable if and only if the degree spectra of R is equal to $\{\mathbf{0}\}$.

Theorem 1.12 (Ash and Nerode 1981, [3]). *A computable model is computably stable (i.e. it is computably isomorphic to any computable copy) if and only if every computable relation on it is intrinsically computable.*

and

Theorem 1.13 (Moses 1983 [41], 1984 [42]). *The computably categorical* 1*-computable linear orderings are precisely those with order type* $\sum_{i=1}^{n}(\mathbf{k}_i+\mathbf{g}_i)+\mathbf{k}_{n+1}$ *where* $\mathbf{k}_i$ *is finite and* $\mathbf{g}_i \in \{\boldsymbol{\omega}, \boldsymbol{\omega}^*, \boldsymbol{\zeta}\} \cup \{\mathbf{d}\cdot\boldsymbol{\eta} :$ $\mathbf{d}$ *is finite*$\}$ *for all* i.

together with

Theorem 1.14 (Moses 1983 [41], 1984 [42]). *A computable linear ordering is* 1*-computable if and only if it has computable successivity relations.*

Similarly, the first and the second can be interwoven. Complexity of a self-embedding of a computable linear ordering are related to that of a *choice set*[24]for it. In fact, the following *Theorem* 1.15 was proved simply by applying *Theorem* 1.16 in [13] (Downey and Moses, 1989)

Theorem 1.15. *Every computable discrete linear ordering — of order type* $\boldsymbol{\zeta}\cdot\tau$ *(with* τ *any order type) — has a recursive copy with no strongly non-trivial* Π^0_1 *self-embedding.*

[24] A choice set for a linear ordering is a subset consisting of precisely one element from each block $c_F(a) = \{b : [a,b] \text{ is finite}\}$ of the linear ordering.

Theorem 1.16. *Every computable discrete linear ordering has a computable copy all of whose choice sets have no infinite* Σ^0_2 *subsets.*

Remark 1.17. Connotational investigation of intrinsic features of mathematical structures is conducted under another theme (beyond the computability theme): the provability theme. In particular, as we previously mentioned, we can deal with the second and the third approach by "stripping the assets" from proof theory; this enterprise relative to reverse mathematics can be found in Downey, Hirschfeldt, Lempp and Solomon (2003 [11]) in relation to linear orderings and Simpson (1999 [58]) more comprehensively.

Notation. We reserve script letters $\mathscr{A}, \mathscr{B}, \ldots, \mathscr{L}, \mathscr{M}, \ldots, \mathscr{P}, \mathscr{Q}, \ldots$ for orderings. The *order type* of a linear ordering is the representative of the equivalence class of it, and lowercase Greek letters $\rho, \sigma, \tau, \ldots$ are used for these representatives. The order types of the natural numbers, the integers, the rational numbers, the real numbers, and the n-element chain are denoted by $\boldsymbol{\omega}, \boldsymbol{\zeta}, \boldsymbol{\eta}, \boldsymbol{\gamma},$ and $\mathbf{n}$ respectively. τ^* denotes the backwards order type of τ. We say that an order type is *computable* if it has a computable member. Let M and N be disjoint, and let $l \in L$ and $m \in M$. We then define the *sum* $\langle L, <_L \rangle + \langle M, <_M \rangle$ by obtaining the domain $L \cup M$ and by retaining the order relations $<_L$ and $<_M$ but by setting $l < m$. Let $a <_A b$ and $i <_I j$. We then define the *product of* $\langle I, <_I \rangle$ *copies of* $\langle A, <_A \rangle$, $\langle A, <_A \rangle \cdot \langle I, <_I \rangle$, by setting $\langle A, <_A \rangle \cdot \langle I, <_I \rangle = \sum\{A_i : i \in I\}$ (i.e. (a, i) is lexicographically less than (b, j).) We use expressions such as $\sigma + \tau$ and $\sigma \cdot \tau$ for the order types of sums and products respectively.

Cautions: It should be noted that very fine computability-theoretic distinctions are sensitive to the exact form of the definition of ordering. For instance, we may describe an ordering $\mathscr{L}$ in terms of either $\leq_L$ or $<_L$. We then have:

$$(a,b) \in \leq_L \iff (a,b) \in <_L \ \vee (a,b) \in =_L,$$
$$\text{and} \quad (a,b) \in <_L \iff (a,b) \in \leq_L \ \& \ (a,b) \notin =_L .$$

For linear orderings without computability of $=_L$, since $(a,b) \in \leq_L \Leftrightarrow (b,a) \notin <_L$ and $(a,b) \notin \leq_L \Leftrightarrow (b,a) \in <_L$, computable enumerability of one relation only implies co-computable enumerability of the other, even though their Turing degrees are the same. For partial orderings the situation is more complicated. Of course, described in terms of $\leq_L$, if $\leq_L$ is c.e. so will $=_L$ be.

There are interesting consequences of such observations with regard to embeddings of c.e. linear ordering into $\mathbb{Q}$. The situation was described as part of a more general result by Lawrence Feiner:

Theorem 1.18 (Feiner 1967, [18]). *If a linear ordering $\langle D, \leq_L \rangle$ has c.e. $\leq_L$ (is Σ_1-presented), then $\langle D, \leq_L \rangle$ is Δ_2-isomorphic to a co-c.e. subset of $\mathbb{Q}$ with the usual computable relations $<_{\mathbb{Q}}$ and $=_{\mathbb{Q}}$.*

There is a simple constructive proof of this result, whereby one uses the enumeration of $\leq_L$ to progressively map members a, b of D to corresponding rationals r_a and r_b in $\mathbb{Q}$. The only need for adjustment of the subordering of $\mathbb{Q}$ is if we subsequently get $(a,b) \in \leq_L \ \& \ (b,a) \in \leq_L$. In this case, respecting a priority ordering of the mappings of members of D, we select the higher priority r_a or r_b to be the image of both a and b, while discarding the lower priority r_a or r_b from the embedding, along with the associated part of the embedding itself. It is easy to see that the resulting embedding is actually d-c.e. Of course, Feiner has the above theorem in the case of $\leq_L$ is just Δ_2, and the constructive version of this again is not difficult. But for more detailed analysis of the computational character of the embedding when $\leq_L$ occupies some intermediate level of the Ershov hierarchy, one has to incorporate the bounded adjustments arising with those noted in the simple argument above. Details of this appear in [9].

Arising from Feiner's work, some writers (Richard Watnick, Rodney Downey, etc.) use the term "computable linear ordering" to mean "computable subset of $\mathbb{Q}$", but here "computable linear ordering" will mean "computably presented linear ordering" as in [51] (Rosenstein, 1982) for example.

1.3 Priority Arguments

This section aims to introduce the ideas underlying priority arguments; to see their essential role during the course of computable constructions; and to discuss a basic priority argument within the framework of the tree of strategies method. The idea has been a cornerstone of most proofs in computability theory since Friedberg and Mučnik constructivised the Kleene and Post (1954, [34]) construction of incomparable Turing degrees below $\mathbf{0}'$, using a priority setting of *requirements*. All aims are achieved in proving the following basic result for computable linear orderings.

Theorem 3.5 (Page 45; Rosenstein 1982, [51])**.** *There is a computable linear ordering of order type $\boldsymbol{\zeta}$ that is computably rigid (i.e. has no nontrivial computable automorphism).*

If we look at the set of rational numbers $\mathbb{Q}$ effectively, more precisely, we fix a computable 1-1 correspondence between ω and $\mathbb{Q}$ defined by $n \mapsto r_n$, then we can build an order-isomorphism up to which any computably presented linear ordering becomes a computable subset of $\mathbb{Q}$.

Lemma 1.19 (Cantor, an effective version). *An order type τ is computable if and only if there is a computable subset B of $\mathbb{Q}$ of order type τ. The equivalence is computable.*

Sketch Proof. ($\Longrightarrow$) Assume that a computably presented linear ordering $\langle A, <_A\rangle$ is given and B is set to be empty at the beginning of the construction. The basic idea of our construction of the computable isomorphism is that if r_n appears in A at the stage s, then we put some r_m into B with $m \leq n$; it is always possible since $\mathbb{Q}$ is dense.

($\Longleftarrow$) Take an computable list $\{r_n\}_{n\in\omega}$ of B while we fix a computable 1-1 correspondence f between ω and $\mathbb{Q}$, and stipulate that $f(r_m) < f(r_n)$ if and only if $m <_A n$. □

Thus, we will prove this theorem by showing

> *There is a computable subset of A of $\mathbb{Q}$ of order type $\boldsymbol{\zeta}$ that is computably rigid.*

Proof of **Theorem 3.5**. We use stage superscripts for sets to let A^s denote the set of element put into A by the end of stage s. Given that A^0 is the set of integers, we will build a computable subset $A = \bigcup_s A^s$ of $\mathbb{Q}$ in stages sometimes by putting some elements of the interval $(e, e+1)$ with $e \in \omega$, so that A has no nontrivial computable automorphism.

Firstly, we want to ensure that the subset A of $\mathbb{Q}$ has order type $\boldsymbol{\zeta}$. To do this, we typically break such a desired condition into denumerable (i.e. countably infinite) requirements. Here is a list of requirements for every $e \in \omega$

N_e : The interval $(e, e+1)$ of A has in it at most finitely many elements.

To make sure that every nontrivial automorphism of A is not computable, we diagonalise against all possible computable functions for the nontrivial automorphism. In this case, we satisfy the requirements for each $e \in \omega$

$$P_e : \varphi_e \text{ is not an nontrivial automorphism of } A.$$

A detailed plan to meet a single requirement P_e in isolation — which we call *atomic strategy* (or *basic module*) for P_e and is denoted by $\mathcal{M}_e$ — is thus. We wait for a stage at which we find $x \in \omega$ such that $\varphi_{e,s}(x) \downarrow$. If no such x exists, then the requirement will be met since φ_e is not total. Otherwise, we compute $\varphi_{e,s}(x)$ and $\varphi_{e,s}(x+1)$ and if they appear to be adjoining without intervening elements of A^s, then we put certain $x+1$ many rational elements between x and $x+1$ into A^{s+1}. We say that P_e *requires attention* at stage $s+1$ if $\mathcal{M}_e$ sees successive outputs, and that P_e *receives attention* (or *acts*) at stage $s+1$ if $\mathcal{M}_e$ carries out an enumeration of $x+1$ such elements into A^{s+1}.

However if P_e indulges in its action, there may be a potential conflict with N_i for some $i \in \omega$ since if infinitely many P_e simultaneously allowed new points between x and $x+1$, then N_i would not be satisfied. One resolution of this conflict is to allow P_e to act only if $x(e) = e$, so that for each e, requirement N_e has higher priority than P_e.

Finally, we need to ensure that the subset A of $\mathbb{Q}$ that we will construct is computable, so we satisfy the requirements for each $e \in \omega$.

$$R_e : r_i \in A \iff r_i \in A^s \text{ for all } s \in \omega \text{ and } i \leq \langle e, s \rangle$$

To meet these requirements, if P_e acts, then we choose new $e+1$ elements r_j for which $j > \langle e, s\rangle$, so that all three kinds of requirements *cohere*, giving their priority ranking as follows.

$$N_0 \prec P_0 \prec R_0 \prec N_1 \prec P_1 \prec R_1 \prec N_2 \prec P_2 \prec R_2 \prec \cdots$$

Note that the leftmost requirement has the highest priority. We will verify that A is computable in justification of these requirements after describing our construction.

Together with an analysis of required conditions, which consist of requirements, there are typically two more main components of a priority argument: *construction* and *verification*. Construction part provides an algorithm for A and major part of verification is to check every requirement by induction on e.

Construction

Let A^0 be the set of integers. P_e requires attention at stage $s+1$ if $\varphi_{e,s}(e)$ and $\varphi_{e,s}(e+1)$ are both defined, $\varphi_{e,s}(e+1)$ is a successor of $\varphi_{e,s}(e)$, and there is no elements in the interval $(e, e+1)$ of A_s. If this situation happens, then we choose $e+1$ rational numbers in the interval $(e, e+1)$ but not in $\{r_i : i \leq \langle e, s\rangle\}$, and enumerate them in A^s. Set $A^{s+1} = A^s$, and define a computable parameter $r(e, s) = e+1$, which prevent P_i, $i \neq e$, from enumerating $e+1$ elements into A. Otherwise, go to the stage $s+2$, and define $r(e, s) = 0$ which indicates that no element is enumerated at stage $s+1$. In either case, if r_i, $i \leq \langle e, s\rangle$, is not already in A^{s+1}, we place them in $\overline{A}^{s+1}$.

Verification

Lemma 1.20. *The subset A of $\mathbb{Q}$ constructed is computable, and hence for every e, requirement R_e is met.*

Proof. If requirement P_e acts at stage $s+1$, i.e. module $\mathcal{M}_e$ carries out an enumeration with new rational numbers whose indices are greater than $\langle e, s\rangle$, then we place r_i, $i \leq \langle e, s\rangle$, not yet in A^{s+1} into $\overline{A}^{s+1}$. The same goes for no action of P_e. Thus requirement R_e is met since $r_i \in A$ if and only if $r_i \in A^s$ for all $s \in \omega$ and $i \leq \langle e, s\rangle$. □

Lemma 1.21. *Requirement N_e is never injured in the sense that it never happens that infinitely many rational numbers are enumerated in the interval $(e, e+1)$ due to a uniform action of $\{P_e\}_{e\in\omega}$ at stage $s+1$, and hence N_e is met. Furthermore, $r(e,t) = e+1$ for all $t \geq s+1$ if P_e acts at stage $s+1$, and $r(e,t) = 0$ otherwise; i.e. A has order type $\boldsymbol{\zeta}$.*

Proof. Since the interval $(e, e+1)$ is filled with the elements only by the action of single requirement P_e, N_e is not injured. By definition of the computable parameter $r(e,s)$ during the construction, $r(e,t) = e+1$ for all $t \geq s+1$ if P_e acts at stage $s+1$, and $r(e,t) = 0$ otherwise. That implies A has order type $\boldsymbol{\zeta}$. □

Lemma 1.22. *For every e, requirement P_e is met, acts at most once, and $r(e) = \lim_s r(e,s)$ exists.*

Proof. Since every P_e acts on their own distinct interval $(e, e+1)$, it acts at most once and hence $r(e) = \lim_s r(e,s) = e+1$ so long as P_e will never ever act and define $r(e) = \lim_s r(e,s) = 0$. In either case, $r(e) = \lim_s r(e,s)$ exists. Fix some e for which φ_e is not partial, otherwise φ_e would not an automorphism. If $\varphi_e(e)$ and $\varphi_e(e+1)$ are both defined, then there exists a least stage s such that $\varphi_{e,s}(e)$ and $\varphi_{e,s}(e+1)$ are defined in A^s. Note that no points were ever enumerated in $(e, e+1)$ of A^s since $\varphi_{e,t}(e)$ and $\varphi_{e,t}(e+1)$, $t < s$, has not defined due to the minimality of s. We break into two cases. (1) If $\mathcal{M}_e$ sees that $\varphi_{e,s}(e)$ and $\varphi_{e,s}(e+1)$ is not a successive pair, so $\varphi_e(e+1)$ is not the successor of $\varphi_e(e)$ at all, then no points will ever enumerated into the interval $(e, e+1)$. Thus φ is not an automorphism since no element between $\varphi_e(e)$ and $\varphi_e(e+1)$ has its inverse element. (2) Otherwise, P_e acts at stage $s+1$, so by the end of stage $s+1$, we have $|(e, e+1)| = e+1$ and $|\big(\varphi_e(e), \varphi_e(e+1)\big)| = 0$. Remember the number of points in the interval $(e, e+1)$ is preserved by the minimality. So in order for φ_e to not be a nontrivial automorphism, we need to show that $|\big(\varphi_e(e), \varphi_e(e+1)\big)| \neq e+1$. There are three possibilities.

(i) If $\varphi_e(e) = e$ and $\varphi_e(e+1) = e+1$, then φ_e is the trivial automorphism of A^0.

(ii) If $\varphi_e(e) = k$ and $\varphi_e(e+1) = k+1$ with $e \neq k$, then the possible number of points in $\big(\varphi_e(e), \varphi_e(e+1)\big)$ is either 0 (if no relevant action is taken) or k (by the minimality condition if an action exists.)

(iii) If $\varphi_e(e)$ and $\varphi_e(e)$ are not a natural numbers, then $|\big(\varphi_e(e), \varphi_e(e+1)\big)| = 0$.

Therefore, in any case, φ_e is not an nontrivial automorphism of A. □

This completes the proof. □

Now we recast the above proof within the tree of strategies framework. It is understood, according to Soare (1985 [59, p. 56]), that the nature of this approach is to return to the spirit of the *Baire category theorem*, which states that the intersection of a countable number of dense open subsets of any complete metric space is itself dense in that space (and hence nonempty). His assertion was in anticipation of justifying the satisfaction of all kinds of requirements. The basic idea is derived by the fact that every single requirement should have a dense open subset of a Baire space (topological space in which the Baire category theorem holds) as the *minimal environment* for its strategy to succeed.[25]Thus it is important to take an appropriate downward (or upward) tree, which is an ideal of the Baire space, i.e. an initial segment of it which is closed under intersection. Note that trees always grow downwards and are *full* trees, e.g. 2^ω, ω^ω and $(\omega \times \{1,2,3,4\})^\omega$, in what follows. Let us follow and draw his idea in connection with an application of the Baire category theorem to the tree method. All nodes of the tree are given a priority ordering $\prec$[26]defined by

$$\alpha \prec \beta \ \ (\alpha \text{ has higher priority than } \beta) \iff \alpha \subset \beta \ \vee \ \alpha <_L \beta,$$

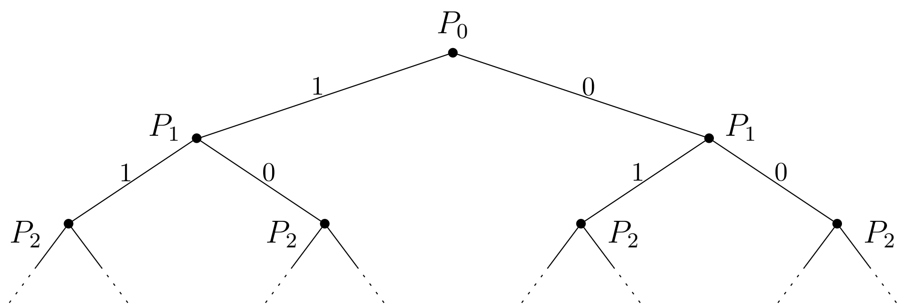
where $\alpha <_L \beta \ \Leftrightarrow \ (\exists a, b \in \Lambda)(\exists \gamma \in T)[\gamma\widehat{\ }\langle a \rangle \subseteq \alpha \ \& \ \gamma\widehat{\ }\langle b \rangle \subseteq \beta \ \& \ a <_\Lambda b]$ and $\langle a \rangle$ and $\langle b \rangle$ mean the string consisting of singletons a and b. Each level of nodes are assigned the atomic strategy of a single requirement and each node is encoded by possible *states* or *outcomes* (mostly in a noneffective way). Now, invoking the Baire category theorem, we intersect all dense open subsets associated with requirements, justifying the existence of a subset of the tree satisfying all the requirements. In the mean time, the priority argument is classified as $\emptyset'$, $\emptyset''$ or $\emptyset'''$ according to how strong oracle of $\emptyset'$, $\emptyset''$ or $\emptyset'''$ is exactly needed to satisfy *each* requirement. In a $\emptyset'$ or $\emptyset''$-priority argument, we can usually form a picture of the *true path* on the tree — as a subset of the tree satisfying all the requirements — at the end of a construction. Visualising the true path is distinctively useful feature of the construction processed in a tree of strategies, for instance, in the case of the $\emptyset'''$-priority argument, it can be

[25]Soare kept his distance from another topological approach of Lachlan (1973 [35]) to the structure of priority arguments.

[26]The original version of this ordering is attributed to Kleene and Brouwer.

viewed as a $\emptyset'$-construction along the true path of a $\emptyset''$-construction (Lachlan 1975, [36]) or as a $\emptyset''$-construction along the true path of a $\emptyset'$-construction (Shore 1988, [57]).

Proof of **Theorem 3.5** *(Tree Proof).* It suffices to meet the same requirements as in the previous proof. Take a tree $T = \Lambda^{<\omega}$ with $\Lambda = \{0, 1\}$, so that our tree is a Baire space. For each $\alpha \in T$, we develop α-*strategy* which is a special version of the basic module for R_e with $|\alpha| = e$ such that α *guesses* that if $\beta = \alpha \upharpoonright k$ and $k < |\alpha|$ then $\beta(k) = \alpha(k)$. For example, β will act only if $\alpha(k) = 1$. This is the idea to assign to α the requirement $P_\alpha = P_e$.



The priority ordering $\prec$ of the nodes on the tree is given as follows

$$\sigma \prec \tau \iff \sigma \subset \tau \ \vee \ (\exists x)[\sigma(x) > \tau(x) \ \& \ (\forall y < x)(\sigma(y) = \tau(y))].$$

For example, $101 \prec 1011$ and $101 \prec 100$. We allow that α *requires attention* at stage $s + 1$ if requirement P_e requires attention and α *receives attention* (or *acts*) at stage $s + 1$ if α's guess *seems correct*, i.e. only when α guesses at the current stage $s + 1$. Eventually we then get a $\emptyset'$ tree strings α of which is encoded as 1 if $|\alpha| = e$ and requirement P_e is *permanently* active after some stage, or else as 0. In other words, we define the outcomes $\{0, 1\}$ on behalf of P_e's *final* action and the $\emptyset'$ tree $\subset \Lambda^{<\omega}$ of outcomes. This tree is what we call *true path.*

Construction

Let A^0 be the set of integers. If α requires attention at stage $s+1$, $|\alpha| = e$, and α's guess seems correct, then we choose $e+1$ rational numbers in the interval $(e, e+1)$ but not in $\{r_i : i \leq \langle e, s\rangle\}$, and enumerate them in A^s. Set $A^{s+1} = A^s$, and define a computable parameter $r(\alpha, s) = r(e, s) = e+1$. Otherwise, go to the stage $s+2$, and define $r(\alpha, s) = r(e, s) = 0$. In either case, if r_i, $i \leq \langle e, s\rangle$, is not already in A^{s+1}, we place them in $\overline{A}^{s+1}$. Now the computable sequence of strings $\{\delta_s : s \in \omega\}$ is defined by $\delta_s(e) = \mathrm{sg} \circ r(e, s)$, which approximates the true path f, so that $f(n) = \lim_{s\to\infty} \delta_s(n)$, where sg is the signum function.

Verification

Lemma 1.23. *The subset A of $\mathbb{Q}$ constructed is computable, and hence for every e, requirement R_e is met.*

Lemma 1.24. *Requirement N_e is never injured in the sense that it never happens that infinitely many rational numbers are enumerated in the interval $(e, e+1)$ due to a uniform action of $\{P_\alpha\}_{\alpha\subset\delta_s}$ at stage $s+1$, and hence N_e is met. Furthermore, $r(\alpha, t) = e+1$ for all $t \geq s+1$ if P_α acts at stage $s+1$, and $r(\alpha, t) = 0$ otherwise; i.e. A has order type $\boldsymbol{\zeta}$.*

Lemma 1.25. *For every e, requirement P_e is met by $\alpha = f \upharpoonright e$, acts at most once, and $r(\alpha) = \lim_s r(\alpha, s)$ exists.*

The proofs of **Lemma 1.23, 1.24 and 1.25** are virtually the same to those of **Lemma 1.20, 1.21 and 1.22** respectively except that the role of P_e is shared by P_α with $|\alpha| = e$ and $\alpha \subset \delta_s$. $\square$

The latter tree proof of the simplest priority argument as in our example does not profit from the guesses of the nodes α, i.e. information for the higher priority requirements P_i, $i < |\alpha|$. The worst of it is that it is not neater than the former one. However, Friedberg-Mučnik theorem previously mentioned benefits from the guesses, so its tree version of the proof should be neater than the original one. In more complicated proofs, there may be a need of incorporations between nodes of strategies in obedience to *Harrington's golden rule*. But this subject is beyond the scope of this thesis.

1.4 The Ershov Hierarchy

In this section, we introduce a hierarchy which intrinsically characterises Δ^0_2-definable sets (or sets Turing reducible to $\emptyset'$)[27], which we call the *Ershov hierarchy*. This hierarchy is obtained by looking at the details of how one approximates Δ^0_2 sets. Note that all the Δ^0_2 sets can be approximated by transfinitely extending the finite level of the Ershov hierarchy. To achieve this, we then introduce Kleene's system of ordinal notations, which we call Kleene's $\mathcal{O}$. Historically, the Ershov hierarchy of finite levels was first introduced by Hillary Putnam (1965 [49]) and Mark Gold (1965 [23]). Later, Yu. Ershov (1968 [15], 1968 [16], 1970 [17]) extended the hierarchy to transfinite levels. A recent and comprehensive article on the Ershov hierarchy is by Marat Arslanov (2011, [1]), and a concise introduction to this hierarchy is found in [67] (Stephan, Yang and Yu, 2009). For details and further background on ordinal notations we refer to Kleene (1955 [31]), Rogers (1967 [50]), Sacks (1990 [53]), Ash and Knight (2000 [2]).

[27] Post's Theorem

It was Joseph Shoenfield who first approximated Δ^0_2 (characteristic) sets in a *limit computable* way.

Lemma 1.26 (Limit Lemma, Shoenfield 1959, [56]). *A set A is Δ^0_2 if and only if there is a computable binary function g such that for all $n \in \omega$, there are cofinitely many stages s at which $\chi_A(n) = g(n,s)$, namely*

$$\lim_{s\to\infty} g(n,s) \textit{ exists and is equal to } \chi_A(n).$$

We now define Kleene's $\mathcal{O}$. Note that the following succinct presentation of Kleene's $\mathcal{O}$ and the Ershov hierarchy is in large part due to that of Stephan, Yang and Yu (2009 [67]).

Definition 1.27 (Kleene 1938, [29]). We define a set of notations $\mathcal{O} \subseteq \omega$, a partial function $|\cdot|_{\mathcal{O}}$ mapping each $a \in \mathcal{O}$ to an ordinal $\alpha = |a|_{\mathcal{O}}$ and a strict partial ordering $<_{\mathcal{O}}$ on $\mathcal{O}$ simultaneously.

- Define 1 to be the notation for 0. In other words, $|1|_{\mathcal{O}} = 0$.
- If a is a notation for α, define 2^a to be a notation for $\alpha + 1$. In other words, $|2^a|_{\mathcal{O}} = \alpha + 1$. Define $b <_{\mathcal{O}} 2^a$ if $b <_{\mathcal{O}} a$ or $b = a$.
- If φ_e is a total computable function such that, for every $n \in \omega$, we have already defined $|\varphi_e(n)|_{\mathcal{O}} = \alpha_n$ and $\varphi_e(n) <_{\mathcal{O}} \varphi_e(n+1)$, then define $|3 \cdot 5^e|_{\mathcal{O}} = \alpha$. Define $b <_{\mathcal{O}} 3 \cdot 5^e$ if there exists some n such that $b <_{\mathcal{O}} \varphi_e(n)$.

This completes the definition of Kleene's system of notations.

We note here that if ordinal $\alpha < \omega$ (i.e. $\alpha = n$ say) then there is a unique $a \in \mathcal{O}$ such that $|a|_{\mathcal{O}} = \alpha$. On the other hand, for any $\alpha \geq \omega$, either $\{a : |a|_{\mathcal{O}} = \alpha\}$ is infinite (if α is constructive as defined below) or empty.

We now give a brief overview of Kleene's system with regard to the notion of a computable ordinal.

Definition 1.28. An ordinal α is defined to be ***constructive*** if for some $a \in \mathcal{O}$, $|a|_{\mathcal{O}} = \alpha$.

Definition 1.29. An ordinal α is defined to be ***computable*** if it is finite or if it is isomorphic to some computable well ordering of ω.

From the following **Theorem 1.30** below, we can deduce that every constructive ordinal is computable.

Theorem 1.30 (Kleene)**.** *There exist computable functions p and q such that for all $b \in \mathcal{O}$,*

(1) $W_{p(b)} = \{a : a <_{\mathcal{O}} b\}$,

(2) $W_{q(b)} = \{\langle u, v\rangle : u <_{\mathcal{O}} v <_{\mathcal{O}} b\}$, *where $\{W_e\}_{e\in\omega}$ is the standard listing of c.e. sets.*

However, we also know that the opposite implication holds.

Theorem 1.31 (Markwald 1954, [39])**.** *Every computable ordinal is constructive.*

Thus, the two notions are equivalent.

Proposition 1.32. *An ordinal α is constructive if and only if it is computable.*

Returning to our main theme, we are already in a position to define the Ershov hierarchy.

Definition 1.33. For each $a \in \mathcal{O}$, a set $A \subseteq \omega$ is defined to be a-c.e. if there are computable functions $f : \omega \times \omega \to \{0,1\}$ and $o : \omega \times \omega \to \mathcal{O}$ such that

(1) For all n, $f(n,0) = 0$ and $o(n,0) <_{\mathcal{O}} a$.

(2) For all n and s, $o(n,s+1) \leq_{\mathcal{O}} o(n,s)$.

(3) For all n and s, if $f(n,s+1) \neq f(n,s)$ then $o(n,s+1) \neq o(n,s)$.

(4) For all n, $\lim_{s\to\infty} f(n,s) = A(n)$.

We use Σ_a^{-1} to denote the class of a-c.e. sets.

Now, roughly speaking, we know that all Δ_2^0 sets appear at level ω^2 of the Ershov hierarchy.

Theorem 1.34 (Ershov)**.** *Every Δ_2^0 set A is a-c.e. for some $a \in \mathcal{O}$ such that $|a|_{\mathcal{O}} = \omega^2$.*

On the other hand, the notations for any given $\alpha < \omega^2$ define a unique class with respect to the Ershov hierarchy in the following sense.

Lemma 1.35 (Ershov)**.** *If $a, b \in \mathcal{O}$ and $|a|_{\mathcal{O}} = |b|_{\mathcal{O}} = \alpha < \omega^2$, then $\Sigma_a^{-1} = \Sigma_b^{-1}$.*

Note 1.36. With **Lemma 1.35** in mind, if $\alpha < \omega^2$, a is a notation for α, and the set $A \in \Sigma_a^{-1}$, we may also say that A is α-c.e. and we may use the (unique) notation Σ_α^{-1} in place of Σ_a^{-1} provided that the context is unambiguous.

Remark 1.37. We remind the reader that the Ershov hierarchy up to level ω is more commonly described by defining a Δ_2^0 set $A \subseteq \omega$ to be ω-c.e. (n-c.e.) if A satisfies for all $n \in \omega$,

$$|\{s : f(n,s+1) \neq f(n,s)\}| \leq g(n), \tag{1.1}$$

where g is a computable function mapping $\omega \to \omega$ (mapping $\omega \to \{n\}$). Note however that on the finite levels of the hierarchy this terminology is not entirely consistent with that derived from **Definition 1.33** and **Note 1.36**. Consider $4 \in \mathcal{O}$ for example. Then $|4|_{\mathcal{O}}$ = the ordinal 2. From **Definition 1.33**, we can deduce that $\Sigma_4^{-1} = \Sigma_1^0$ (the class of c.e. sets). Accordingly, the notation of **Note 1.36** gives, for the ordinal 2, $\Sigma_2^{-1} = \Sigma_1^0$. However, in the context specified by 1.1 above (with $1 \in \omega$) the class of 1-c.e. sets $= \Sigma_1^0$. Therefore, to avoid confusion, we assume that the terminology used below corresponds to that of **Definition 1.33** or otherwise, if specified in terms of ordinals, to that of **Note 1.36**.

With **Section 3.2** in **Chapter 3** (on uniform Δ_2^0 classes) in mind, we now extend our previous terminology relative to the Ershov hierarchy.

Definition 1.38. Given a set $\mathcal{C} \subseteq \mathcal{O}$, we define a set $A \subseteq \omega$ to be $\mathcal{C}$***-c.e.*** if A is a-c.e. for some $a \in \mathcal{C}$ and we use $\Sigma_{\mathcal{C}}^{-1}$ to denote the class of $\mathcal{C}$-c.e. sets. (So that $\Sigma_{\mathcal{C}}^{-1} = \bigcup_{a \in \mathcal{C}} \Sigma_a^{-1}$ by definition.)

In particular, we also extend our terminology to the context of classes of functions.

Definition 1.39. For $a \in \mathcal{O}$, we say that a function (possibly partial) $f : \omega \to \omega$ is a***-c.e.*** if $G(f)$ is a-c.e. and we use the notation $f \in \Sigma_a^{-1}$ in this case. We also extend this notation to subsets $\mathcal{A} \subseteq \mathcal{O}$ and to ordinals $\alpha < \omega^2$ in the way described above.

Note that we could define a function g to be *argument a-c.e.* (*α-c.e.* if $\alpha < \omega^2$) if we replace $\{0, 1\}$ by ω and A by g in **Definition 1.33**. This notion — which gives a measure of how many times the approximation to a function "changes its mind" on each argument — is at most as general as the standard notion given in **Definition 1.39** in the sense that the class of argument a-c.e. (α-c.e.) functions is subsumed by the class of a-c.e. (α-c.e.) functions for any $a \in \mathcal{O}$ ($\alpha < \omega^2$). In fact, we can show, in the case of $\alpha = \omega$, that the former is strictly less general than the latter by showing, using a straightforward diagonalisation argument, that there exists a total Π^0_1 function g (i.e. $G(g) \in \Pi^0_1$) such that g is not argument ω-c.e.

Chapter 2

Linearisations of Computable Partial Orderings

We describe an approach to refining the computable content of what is known about linearisations of countable partial orderings in the Ershov hierarchy, which preserve natural properties of orderings. This is illustrated by positive results to show that any computably well-founded computable partial ordering has an ω-c.e. linear extensions which is computably well-founded. We then positively conjecture that any computably scattered computable partial ordering has an ω-c.e. linear extensions which is computably scattered, and further discuss about reducing the gap between negative and positive results in terms of the Ershov hierarchy in both cases of computable well-foundedness and computable scatteredness.[1]

[1]We acknowledge helpful comments from S. Barry Cooper and Anthony Morphett during the preparation of this chapter. [9]

2.1 Introduction

In 1930, Edward Szpilrajn gave a result of great importance which everything else in the theory of partial orderings depends on.

Theorem 2.1 (Szpilrajn 1930, [68])**.** *Every partial ordering has a linear extension.*

It is well known that the computable version of Szpilrajn's theorem also holds.

Theorem 2.2 (Folklore, see Downey 1998, [10])**.** *Every computable partial ordering has a computable linear extension.*

Robert Bonnet, Maurice Pouzet, Frederick Galvin and Ralph McKenzie (see Bonnet and Pouzet 1982, [6]) developed Szpilrajn's theorem classically by examining the natural question of to what extent such a linearisation may preserve a property P of the ordering — while focussing particularly on commonly encountered properties P such as *well-foundedness* and *scatteredness*. The description they found of the countable suborderings whose avoidance is generally retainable by a suitably chosen linearisation shows that any well-founded partial ordering has a well-founded linearisation; and any scattered partial ordering has a scattered linearisation. Note that a partial ordering $\langle A, <_A \rangle$ is *well-founded* if there is no infinite descending sequence under $<_A$, and $\langle A, <_A \rangle$ is *scattered* if there is no suborderings of A which has order type $\boldsymbol{\eta}$.

Theorem 2.3 (Bonnet 1969, [4])**.** *Every well-founded partial ordering has a well-founded linear extension.*

Theorem 2.4 (Bonnet and Pouzet 1969, [5], and (independently) Galvin and McKenzie)**.** *Every scattered partial ordering has a scattered linear extension.*

The proofs of **Theorem 2.3 and 2.4** depend on non-constructive ingredients. So computationally informative counterparts of these results may be obtainable, or not, according to the computational constraints applied. Kierstead and Rosenstein gave a semi-effective version of this result.

Theorem 2.5 (Kierstead and Rosenstein 1984, [52])**.** *Every well-founded computable partial ordering has a well-founded computable linear extension.*

To get a fully effective version, weaker notions of well-foundedness and scatteredness was introduced by Rosenstein: An ordering $<_A$ is *computably well-founded* if there is no infinite computable sequence which is decreasing under $<_A$, and $\langle A, <_A\rangle$ is *computably scattered* if there is no computable suborderings of A which has order type η. Rosenstein showed that **Theorem 2.5** fails for computable well-foundedness.

Theorem 2.6 (Rosenstein 1984, [52])**.** *There is a computably well-founded computable partial ordering with no computably well-founded computable linear extension.*

Rosenstein's counter-example is a computable tree T with no computable paths; given any computable linear extension $<_B$ of T, a "$<_B$-first search" through the tree T yields a computable infinite descending sequence.

On the other hand, in the case of computable scatteredness, Downey, Hirschfeldt, Lempp and Solomon studied[2]the proof-theoretic strength of **Theorem**

[2]They also studied the proof-theoretic strength of **Theorem 2.3**.

Theorem 2.7 (Downey, Hirschfeldt, Lempp and Solomon 2003, [11])**.** *(1) "Every well-founded partial ordering has a well-founded linear extension" is provable in ACA_0.*

(2) "Every well-founded partial ordering has a well-founded linear extension" proves WKL_0 over RCA_0.

(3) "Every well-founded partial ordering has a well-founded linear extension" is not provable in WKL_0.

However, these results are not orientated towards our study on computability theoretic complexity of well-founded linear extensions in the Ershov hierarchy.

2.4 in the spirit of reverse mathematics. We do not go into detail but give their results.

Theorem 2.8 (Downey, Hirschfeldt, Lempp and Solomon 2003, [11]). *(1) "Every scattered partial ordering has a scattered linear extension" is provable in* Π^1_1-CA_0. *(Independently proved by Howard Becker)*

(2) "Every scattered partial ordering has a scattered linear extension" is not provable in WKL_0.

The point is that their poof gave a negative answer for computable scatteredness.

Theorem 2.9 (Downey, Hirschfeldt, Lempp and Solomon 2003, [11]). *There is a clasically scattered, computable partial ordering such that every computable linear extension has a computable densely ordered subchain.*

Rosenstein did however give a bound on the computational complexity necessary to obtain a computably well-founded (computably scattered) linear extension of a computably well-founded (computably scattered) computable partial ordering.

Theorem 2.10 (Rosenstein 1984, [52]). *Every computably well-founded computable partial ordering has a computably well-founded* Δ^0_2 *linear extension.*

Theorem 2.11 (Rosenstein 1984, [52]). *Every computably scattered computable partial ordering has a computably scattered* Δ^0_2 *linear extension.*

Rosenstein's proof of **Theorem 2.10** uses an oracle construction with a $\emptyset'$ oracle. However, it is not clear from Rosenstein's construction whether there is a computable bound on the number of oracle queries. By rephrasing Rosenstein's construction as a full-approximation priority argument, we improve the complexity of the linear extension from Δ^0_2 to ω-c.e.

Conventions. In what follows, we identify a partial ordering $\langle A, <_A\rangle$ with its order relation $<_A$ since the complexity of $\langle A, <_A\rangle$ is measured as that of $<_A$. Let D be a subset of ω. We define a *partial ordering* on D to be a relation $<_A$ on $D \times D$ satisfying

(i) $a \not<_A a$ for all $a \in D$ (irreflexivity),

(ii) if $a <_A b$ then $b \not<_A a$ (asymmetry),

(iii) if $a <_A b$ and $b <_A c$ then $a <_A c$ (transitivity).

If neither $a <_A b$ nor $b <_A a$ then we write $a|_A b$. Say that a and b are *comparable* if $a <_A b$ or $b <_A a$; otherwise they are *incomparable*.

We sometimes regard the partial ordering $<_A$ as a function r_A from $D \times D$ to the set of symbols $\{<_A, >_A, =_A, |_A\}$ in the natural way, where $r_A(a, b) = =_A$ if and only if $a = b$. The relation $<_A$ is computable (respectively Δ^0_2, ω-c.e., etc.) if the function r_A is computable (Δ^0_2, ω-c.e., etc.); obviously, every finite ordering is computable.

A *linear ordering* is a partial ordering $<_B$ such that for every a, b ($a \neq b$) either $a <_B b$ or $b <_B a$. The linear ordering $<_B$ is an *extension* of a partial ordering $<_A$ if $a <_A b$ implies $a <_B b$, i.e. $<_A$ and $<_B$ agree on all $<_A$-comparable elements. It is convenient to consider $<_B$ as an extension of $<_A$ even if the domain of $<_B$ is strictly larger than that of $<_A$.

We will also look at a partial ordering $<_A$ as a set of axioms

$$S_A = \{\langle a, b\rangle \in \omega : a <_A b\}.$$

An *axiom* is a pair $\langle a, b\rangle$ asserting that $a <_A b$. A partial ordering $<_B$ is an extension of $<_A$ if $S_A \subseteq S_B$. Let S_* be a set of axioms satisfying irreflexivity (i) and asymmetry (ii) as well as

$$\nexists a, b, c \text{ such that } a <_* b,\ \ b <_* c \text{ and } c <_* a. \tag{2.1}$$

Although $<_*$ may not be an ordering because transitivity (iii) may fail, we can extend $<_*$ to a partial ordering $<_B$ by taking the *transitive closure*: the least partial ordering $<_B$ (by extension) such that $S_* \subseteq S_B$. That is, for any a, b, c such that $a <_* b$ and $b <_* c$, we add to $<_B$ the axiom that $a <_B c$. If X is a subset of the domain D of $<_A$, we denote the restriction of $<_A$ to X by $<_A \restriction X$, which is the ordering given by

$$S_{A\restriction X} = \{\langle a, b\rangle \in S_A : a, b \in X\},$$

i.e. obtained from $<_A$ by discarding any axioms involving numbers not in X.

2.2 Computably Well-Founded ω-c.e. Linear Extension

Theorem 2.12. *Every computably well-founded computable partial ordering $<_A$ (with domain ω) has a computably well-founded ω-c.e. linear extension $<_B$ (with domain ω).*

Proof. Let $<_A$ be a computably well-founded computable partial ordering. We will build a uniformly[3]computable sequence $\{<_{B,s}\}_{s\geq 0}$ of finite linear orderings

[3]Let D_y denote finite set with index $y \in \omega$ in some canonical listing of all finite subsets of ω. Then there is a computable function f such that $<_{B,s}= D_{f(s)}$ (as a set of axioms).

such that the limit

$$<_B = \lim_{s\to\infty} <_{B,s}$$

exists, has domain ω and is a linear extension of $<_A$. By examining the construction, we shall be able to read off a computable bound for the number of changes in $<_{B,s}$ pointwise, and hence we can observe that the limit $<_B$ is ω-c.e. More precisely, let $r_s : \omega \times \omega \to \{<, >, =, |\}$ be the function (uniformly computable in s) such that $r_s(a, b)$ agrees with $<_{B,s}$ if a, b are in the domain of $<_{B,s}$, and $r_s(a, b) = |$ if a or b are not in the domain of $<_{B,s}$. We will ensure that the *change set*

$$\{s : r_s(a, b) \neq r_{s+1}(a, b)\}$$

is bounded by some computable function in a and b and hence $<_B$ $(= \lim_s r_s)$ is ω-c.e.

Let $\{W_e\}_{e\in\omega}$ be a standard listing of all c.e. sets. Let $x^e_0, x^e_1, \ldots$ be the elements of W_e in the order that they are enumerated into W_e. If W_e is finite, then only finitely many x^e_i are defined. Say that x^e_i is defined at stage s if at least $i + 1$ many numbers have been enumerated into W_e by stage s; otherwise x^e_i is undefined at s. If W_e is infinite, then each x^e_i is eventually defined and the sequence $(x^e_i)_{i\in\omega}$ is infinite. Note that for any computable sequence $(z_i)_{i\in\omega}$, there is some e such that $z_i = x^e_i$.

To make $<_B$ computably well-founded, we will ensure that each sequence $(x^e_i)_{i\in\omega}$ does not give an infinite descending sequence under $<_B$. The basic strategy to achieve this (for a fixed e, and dropping the e superscript) is to look for x_i, x_j with $i < j$ and $x_i|_A x_j$ (or $x_i <_A x_j$). When we find such x_i, x_j, we define $x_i <_B x_j$. As long as no other requirement later changes $x_i >_B x_j$, then we will succeed in ensuring that $(x_i)_{i\in\omega}$ is not a descending sequence under $<_B$. If the sequence (x_i) is infinite, then we must eventually find a suitable x_i, x_j, as otherwise (x_i) would be an infinite descending sequence under $<_A$, which is impossible since $<_A$ is computably well-founded.

2.2.1 Requirements

The construction will satisfy the following requirements for $e \in \omega$,

N : $<_B$ is a linear extension of $<_A$.

P_e : If the sequence $(x_i^e)_{i\in\omega}$ is infinite, then there is some i, j with $i < j$ and $x_i^e <_B x_j^e$.

To satisfy all the requirements, we place them in a *finite injury construction*, ordering the requirements in the priority ordering

$$N \prec P_0 \prec P_1 \prec P_2 \prec \cdots$$

2.2.2 Strategies

The Strategy for N

To ensure that $<_B$ is a linear extension of $<_A$, at every stage we will define $<_{B,s}$ to be a linear extension of $<_A\restriction s$.

To ensure that $\lim_{s\to\infty} <_{B,s}$ exists (and is in fact ω-c.e.), we will not allow requirement P_e to modify $<_{B,s}\restriction e$. Since only finitely many requirements are allowed to modify $<_{B,s}\restriction e$ during the course of the construction, and as we will argue that each requirement acts only finitely often, the limit $\lim_{s\to\infty} <_{B,s}$ exists (and has a computable bound on its changes).

The Strategy for P_e

Each requirement P_e has a threshold $l_e[s]$ which is the portion of $<_B$ that P_e wishes to preserve to ensure that P_e remains satisfied. We will explicitly set l_e during the construction; $l_e[s]$ denotes the value of l_e at the beginning of stage s. Let

$$L_e[s] = \{l_{e'}[s] : e' < e\};$$

this is the portion of $<_B$ that is restrained by higher-priority requirements and that P_e is not permitted to modify. So the role of $L_e[s]$ is to prevent P_e from injuring higher-priority P requirements.

The basic strategy for P_e can be summarised as follows.

(i) Look for x_i^e, x_j^e $(\in \omega^{[e]})$[4]such that $i < j$, $x_i^e|_A x_j^e$, and such that we could define $x_i^e <_{B,s+1} x_j^e$ without affecting $<_{B,s+1}\restriction L_e[s]$.

(ii) When such x_i^e, x_j^e are found, define $x_i^e <_{B,s+1} x_j^e$ and set $l_e[s+1] = \max(x_i^e, x_j^e)$.

(iii) Define the rest of $<_{B,s+1}$ in such a way to preserve $<_{B,s+1}\restriction L_e[s] =<_{B,s}\restriction L_e[s]$ and to make $<_{B,s+1}$ be a linear extension of $<_A\restriction s+1$.

The schematically outlined *momentous process* of the construction achieved by (ii) and (iii) consists of (ii) finite extension verified by diagonalisation (simply called diagonalisation) first (giving $<_*$) and (iii) taking transitive closure plus applying a computable version of Szpilrajn's linearisation next (giving $<_B^{s+1}$). Note that the construction of the computable version of Szpilrajn's linearisation consists of defining axioms and then taking transitive closure.

[4] $\omega^{[e]} = \{\langle x, e\rangle : x \in \omega\}$. Note that $x_i^e, x_j^e \geq e$, which ensures that $<_B$ is ω-c.e.

$$\underbrace{\text{Diagonalisation}}_{\text{Defining Axioms}} \rightarrow \text{Taking Transitive Closure}$$

$$\rightarrow \underbrace{\text{Computable Version of Szpilrajn's Linearisation}}_{\text{Defining Axioms} \rightarrow \text{Taking Transitive Closure}}$$

And we will argue that if the sequence (x_i^e) is infinite then we will eventually find a suitable x_i^e, x_j^e and will not get stuck waiting at (i).

Say that P_e *requires attention* at stage $s+1$ if

(I) [P_e *is not yet satisfied*] there does not exist $i' < j'$ such that $x_{i'}^e, x_{j'}^e$ are defined at s, $x_{i'}^e, x_{j'}^e \leq \max(L_e[s], l_e[s])$ and $x_{i'}^e <_{B,s} x_{j'}^e$;

(II) [*we can satisfy* P_e *by setting* $x_i^e <_B x_j^e$] there is $i < j$ such that x_i^e, x_j^e are defined at s and there is a linear extension $<_*$ of

$$\underbrace{<_{B,s}\restriction L_e[s] \quad \cup \quad <_A\restriction \max(x_i^e, x_j^e, s)}_{\text{the transitive closure of the set of axioms}}$$

such that $x_i^e <_* x_j^e$.

2.2.3 Construction

Initially, set $r_0(x,y) = |$ for all $x, y \in \omega$, and $l_e[0] = 0$ for all e.

At the beginning of stage $s+1$, let e be the least such that P_e requires attention at stage $s+1$.

Action. If there is such an e, let i, j and $<_*$ be as in (II) and set $<_{B,s+1} = <_*$, $l_e[s+1] = \max\{x_i^e, x_j^e\}$ and $l_{e'}[s+1] = 0$ for all $e' > e$, saying that P_e *acts* at

$s+1$. If there is no such e, then let $<_*$ be a linear extension of

$$\underbrace{<_{B,s} \ \cup \ <_A \restriction s}_{\text{the transitive closure of the set of axioms}}$$

and set $<_{B,s+1} = <_*$ (if $<_B$ already has domain larger than s, then we can just set $<_{B,s+1} = <_{B,s}$). Note that such $<_*$ exists by Szpilrajn's theorem.

2.2.4 Verification

Lemma 2.13. *$<_B$ is ω-c.e. (if it exists. See* **Lemma 2.14***.)*

Proof. If P_e acts at s, it remains satisfied at all stages after s unless perhaps some stronger priority requirement acts after s. Therefore, P_e can act at most 2^e times. Since P_e can never modify $<_{B,s}$ on numbers $\leq e$, $<_{B,s}\restriction e$ can change at most $\sum_{i<e} 2^i = 2^e - 1$ times. Therefore, $<_B$ is ω-c.e. □

Lemma 2.14. ***(The construction is finitary in the sense of compactness, for example according to Kőnig's lemma.)*** *Each requirement P_e and N are satisfied. Namely, diagonalisation does not fail even if we take transitive closure — and then by Spizlrajn's theorem, $<_B$ $(= \lim_s <_{B,s})$ defines a linear extension of $<_A$, whose existence is ensured by Kőnig's lemma.*

Proof. If W_e is finite, the result is immediate. Let s_0 be a stage such that $L_e[s]$ is fixed after s_0 — which exists because, as noted above, requirements $R_{e'}$ for $e' < e$ act at most $2^e - 1$ times. It suffices to show that (II) holds for P_e at any $s > s_0$. Notice firstly that the only way that (II) could fail to hold eventually for P_e is if for every such x_i^e, x_j^e and $>_*$ as in (II), we always have $x_i^e >_* x_j^e$

because of the transitive closure of $<_{B,s}\upharpoonright L_e[s_0]$. That is, there is always an $a, b \leq L_e[s_0]$ such that $x^e_j <_A a <_{B,s_0} b <_A x^e_i$. We argue that this cannot occur.

Fix e and suppose that W_e is infinite. Define an equivalence relation $\sim$ on numbers greater than $L_e[s_0]$ as follows: $x \sim y$ if for all $a \leq L_e[s_0]$ we have

$$x <_A a \iff y <_A a,\ x >_A a \iff y >_A a \text{ and } x|_A a \iff y|_A a.$$

Since W_e is infinite, some equivalence class I is infinite. Then $W_e \cap I$ is an infinite c.e. set; let $x_{k_0}, x_{k_1}, \ldots$ (dropping the subscript e) be the subsequence of $x_0, x_1, \ldots$ consisting of the elements of $W_e \cap I$ in the order that they are enumerated into W_e. Since $<_A$ is computably well-founded, the sequence $x_{k_0}, x_{k_1}, \ldots$ cannot be an infinite descending sequence under $<_A$. Therefore, there are x_{k_i}, x_{k_j} $(k_i < k_j)$ with $x_{k_i}|_A x_{k_j}$ or $x_{k_i} <_A x_{k_j}$. Therefore, our construction will be made safely in the sense that the transitive closure of $<_{B,s_0} \cup <_A\upharpoonright L_e[s_0]$ will be a finite partial ordering with $x_{k_i}|x_{k_j}$ and then by Szpilrajn's theorem, there will be a finite linear extension $<_*$ with $x_{k_i} <_* x_{k_j}$. Note that our limit $<_B$ exists by Kőnig's lemma since $W_e \cap I$ is infinite. □

Lemma 2.15. ***(The game to construct*** $<_B$ ***has a winning strategy, i.e., is determinate.)*** *$<_B$ is computably well-founded. Namely, each requirement P_e is satisfied — and hence, a computable version of Baire Category Theorem ensures $<_B$ meets every requirement.*

Proof. First, fix e and suppose that W_e is infinite. Suppose that $x_i >_A x_j$ or $x_i|_A x_j$ (dropping the superscript e) for all i, j. Note that there are infinitely many i, j such that $x_i|_A x_j$; if not, we would have $x_i >_A x_{i+1}$ for all sufficiently large i, which would be a computable infinite descending sequence under $<_A$, which is impossible since $<_A$ is computably well-founded. If (II) holds for P_e

at any $s > s_0$ then P_e will be permanently satisfied. But (II) holds by **lemma 2.14**. Now, applying a computable version of Baire category theorem, we know that $<_B$ meets every requirement and hence is computably well-founded. □

This completes the proof of **Theorem 2.12**. □

2.3 Open Questions

In this section, we pose an open problem on whether the property "computably scattered" is preserved, and mention possibilities for improving the results in terms of the Ershov hierarchy in both cases of computable well-foundedness and computable scatteredness by giving positive conjectures.

Conjecture 2.16. *Every computably scattered computable partial ordering has a computably scattered ω-c.e. linear extension.*

There are clearly plausible approaches to verifying this conjecture. The details are more complicated than those for the computably well-founded case due to the higher logical complexity of the property of scatteredness.

Conjecture 2.17. *Every computably well-founded computable partial ordering has a computably well-founded d-c.e. linear extension.*

For this, one does have a strategy for proving the result, which non-trivially extends the basic approach of the ω-c.e. results. This would provide a complete solution to **Question 6.4** in Downey (1998 [10]).

Conjecture 2.18. *Every computably scattered computable partial ordering has a computably scattered d-c.e. linear extension.*

And for this conjecture, we have no specific strategy outlined.

Note that **Conjecture 2.17** (**Conjecture 2.18**) implies **Theorem 2.12** (respectively, **Conjecture 2.16**).

Chapter 3

Automorphisms of Computable Linear Orderings

We define the property "*uniform* Δ_2^0" relative to classes of functions from ω to ω and we show that the class of a-c.e. functions, $a \in \mathcal{O}$, has this property. We show, for example, that for any graph subuniform Δ_2^0 class $\mathcal{F}$ there exist computable linear orderings of order type $\mathbf{2} \cdot \boldsymbol{\eta}$ and $\boldsymbol{\omega} + \boldsymbol{\zeta}$ which are $\mathcal{F}$-*rigid* (see **Definition 3.7**) and we discuss about generalisations of these results.[1]

3.1 Introduction

In 1940, Ben Dushnik and Edwin Miller gave the existence of non-trivial self-embeddings (i.e. non-identity order-preserving 1-1 mappings the domain and range of which are the same) of a denumerable linear ordering.

[1]We have benefitted from useful advice from S. Barry Cooper and Charles M. Harris during the final presentation and this material. [8]

Theorem 3.1 (Dushnik and Miller 1940, [14]). *Every denumerable linear ordering has a non-trivial self-embedding.*

In particular, the result for denumerable linear orderings which have the intervals of order type $\boldsymbol{\omega}$ or $\boldsymbol{\omega}^*$ follows from the initial part of their argument by mapping an element in the interval of order type $\boldsymbol{\omega}$ ($\boldsymbol{\omega}^*$) to the immediate successor (the immediate predecessor).

Corollary 3.2 (Dushnik and Miller 1940, [14]). *There is a linear ordering of order type $\boldsymbol{\omega}$ (or $\boldsymbol{\omega}^*$) which has no nontrivial self-embedding.*

It however turned out that **Theorem 3.1** is not effective in the sense of the following.

Theorem 3.3 (Hay and Rosenstein 1982, [51]). *There is a computable linear ordering of order type $\boldsymbol{\omega}$ (or $\boldsymbol{\omega}^*$) which has no nontrivial computable self-embedding.*

Furthermore, the effectiveness of this results was measured by Rodney Downey and Steffen Lempp.

Theorem 3.4 (Downey and Lempp 1999, [12]). *There is a computable linear ordering $\mathscr{L}$ such that if f is a nontrivial self-embedding of $\mathscr{L}$ then f can compute $\emptyset'$.*

On the other hand, it was easily observed by Dushnik and Miller (1940 [14]) that the interval of order type $\boldsymbol{\zeta}$ ($= \boldsymbol{\omega}^* + \boldsymbol{\omega}$) would give a non-trivial automorphism, and Joseph Rosenstein gave an effective version of this result and its complexity. The proofs of these results are similar to those of self-embedding cases.

Theorem 3.5 (Rosenstein 1982, [51]). *There is a computable linear ordering of order type* $\boldsymbol{\zeta}$ *that is computably rigid (i.e. has no nontrivial computable automorphism). In fact, its automorphisms are at best* Π^0_1*-definable.*

It was Steven Schwarz who showed that computable rigidity is characterised by a classical order type.

Theorem 3.6 (Schwarz 1984, [55]). *For every computable linear ordering that is not rigid, it is computably rigid if and only if it contains no interval of order type* $\boldsymbol{\eta}$*.*

Now we generalise the notion of "computable rigidity" to broader classes.

Definition 3.7. For a class of functions $\mathcal{F}$ and a linear ordering $\mathscr{L}$, we say that $\mathscr{L}$ is $\mathcal{F}$***-rigid*** if there exists no nontrivial automorphism f of $\mathscr{L}$ such that $f \in \mathcal{F}$.

This suggests a study of a determination of the level of arithmetical hierarchy at which rigidity for computable linear orderings which contains no interval of order type $\boldsymbol{\eta}$ breaks down, i.e. a classification of $\mathcal{F}$*-rigidity* of such computable linear orderings where $\mathcal{F} \subset \bigcup_{n \geq 0}(\Sigma^0_n \cup \Pi^0_n)$. Order types of those computable linear orderings include $\boldsymbol{\omega}$, $\boldsymbol{\omega}^*$, $\boldsymbol{\omega} + \boldsymbol{\zeta}$, ... , and some $\boldsymbol{\eta}$-like order types (those which have the form $\sum\{f(q) \in \omega - \{0\} : q \in \mathbb{Q}\}$ where f is from $\mathbb{Q}$ to $\omega - \{0\}$) such as $\mathbf{2} \cdot \boldsymbol{\eta}$, $\boldsymbol{\zeta} \cdot \boldsymbol{\eta}$, etc.

For $\boldsymbol{\eta}$-like order types, Henry Kierstead studied the order type $\mathbf{2} \cdot \boldsymbol{\eta}$.

Theorem 3.8 (Kierstead 1987, [27]). *There is a computable linear ordering of order type* $\mathbf{2} \cdot \boldsymbol{\eta}$ *which has no nontrivial* Π^0_1 *automorphism (i.e. is* Π^0_1*-rigid.)*

The same result was proved for $\boldsymbol{\zeta} \cdot \boldsymbol{\eta}$ in [13] (Downey and Moses, 1989). (See **Theorem 1.15** and **1.16** in Chapter 1.) In sum, given that nontrivial automorphisms of computable linear orderings of either of the order types $\mathbf{2} \cdot \boldsymbol{\eta}$ and $\boldsymbol{\zeta} \cdot \boldsymbol{\eta}$ exist, their complexities are Δ^0_2-definable in the arithmetical hierarchy. It is natural to intrinsically refine the Δ^0_2-definable class, for example, in terms of the Ershov hierarchy. In this chapter, we improve Kierstead's result for the order type $\mathbf{2} \cdot \boldsymbol{\eta}$ in terms of such refinements. And we also do this for the order type $\boldsymbol{\omega} + \boldsymbol{\zeta}$.

3.2 Uniform Δ^0_2 Classes

In this section, we introduce uniform Δ^0_2 classes and look at the Ershov hierarchy in terms of this notion. Recall, by the limit lemma, Δ^0_2 (characteristic) sets can be approximated in a *limit computable* way. Broadening our interests to partial functions, we uniformise Δ^0_2 partial functions, and then define *uniform* Δ^0_2 *classes* and *graph uniform* Δ^0_2 *classes.*

Notation. If f is a binary (ternary) function then f_e ($f_{e,s}$) is shorthand for $\lambda n.f(e,n)$ ($\lambda n.f(e,n,s)$).

Definition 3.9. If $\mathcal{F}$ is a class of unary functions (mapping $\omega \to \omega$), $\mathcal{F}$ is defined to be ***uniform*** Δ^0_2 (***subuniform*** Δ^0_2) if there is a binary function $f \leq_T \emptyset'$ such that

$$\mathcal{F} = \{f_e : e \in \omega\} \qquad (\mathcal{F} \subseteq \{f_e : e \in \omega\}).$$

A class of sets $\mathcal{C} \subseteq \mathcal{P}(\omega)$ is defined to be ***uniform*** Δ^0_2 (***subuniform*** Δ^0_2) if the class of characteristic functions of $\mathcal{C}$ is uniform Δ^0_2 (subuniform Δ^0_2). A class $\widehat{\mathcal{F}}$ of (partial) unary functions (mapping $\omega \to \omega$) is defined to be ***graph uniform*** Δ^0_2 (***graph subuniform*** Δ^0_2) if the class $\{G(f) : f \in \widehat{\mathcal{F}}\}$ is uniform Δ^0_2 (subuniform Δ^0_2).

We note here that the notion "uniform Δ^0_2" corresponds to the notion "$\mathbf{0}'$-uniform" derived from Jockusch's notation (1972 [26]). Precisely, he defined it for Turing degrees. Thus, the notion of "uniform Δ^0_2 class of total functions" has the same meaning of that of "$\mathbf{0}'$-uniform class of total functions" simply by applying Post's theorem.

The motivation for the present terminology is due to our use of **Definition 3.10** below.

Definition 3.10. We say that a computable function $f : \omega \times \omega \times \omega \to \omega$ is ***uniform*** Δ^0_2 ***approximating*** if $\lim_{s\to\infty} f_{e,s}(n)$ exists for all $e, n \in \omega$ and, in this case, we say that $\{f_{e,s}\}_{e,s\in\omega}$ is a ***uniform*** Δ^0_2 ***approximation***. Accordingly, f defines a class $\{f_e\}_{e\in\omega}$ such that $f_e(n) = \lim_{s\to\infty} f_{e,s}(n)$ for all $e, n \in \omega$.

Notation. Following standard practice, we use the notation $f(n)\downarrow$ to denote that the function f is defined at argument n. Likewise, we use this notation in the context of computations, for example $\varphi(n)\downarrow$ denotes the convergence of the computation of Turing machine φ with input n. However, for simplicity, we also use this notation for the convergence in the limit (of one argument) for total binary functions. For example, we use "$\lim_{s\to\infty} f_s(n)\downarrow$" as shorthand for "$\lim_{s\to\infty} f_s(n)$ exists". Moreover, we use the shorthand "$\liminf_{s\to\infty} f_s(x) = \infty$" to denote that $\liminf_{s\to\infty} f_s(x)$ tends to infinity.

By application of the limit lemma, we know that **Definition 3.9** can be derived from this notion.

Lemma 3.11. *A class of functions $\mathcal{F}$ is uniform Δ^0_2 if and only if there exists a uniform Δ^0_2 approximation function f such that $\mathcal{F} = \{f_e\}_{e\in\omega}$. In particular, a class of sets $\mathcal{C}$ is uniform Δ^0_2 if and only if there exists a uniform Δ^0_2 approximation $\{A_{e,s}\}_{e,s\in\omega}$ such that $\mathcal{C} = \{A_e\}_{e\in\omega}$. (Notice here our usual identification of a set predicate with its characteristic function.)*

We now introduce some uniform Δ^0_2 classes relevant to the next section.

Definition 3.12. We say that the computable function $f : \omega \times \omega \times \omega \to \omega$ is ***upwards uniform*** Δ^0_2 ***approximating*** if for all $e, x \in \omega$, either

(1) $\lim_{s\to\infty} f_{e,s}(x) \downarrow$, or

(2) $\liminf_{s\to\infty} f_{e,s}(x) = \infty$.

In this case, we say that $\{f_{e,s}\}_{e,s\in\omega}$ is an ***upwards uniform*** Δ^0_2 ***approximation***. Accordingly, f defines a class of partial functions $\{f_e\}_{e\in\omega}$ such that for every index e and all $n \in \omega$, $\mathrm{Dom}(f_e) =_{\mathrm{def}} \{n : \lim_{s\to\infty} f_{e,s}(n) \downarrow\}$ and such that for every $n \in \mathrm{Dom}(f_e)$, $f_e(n) =_{\mathrm{def}} \lim_{s\to\infty} f_{e,s}(n)$. We say that the class $\{f_e\}_{e\in\omega}$ is ***upwards uniform*** Δ^0_2.

Lemma 3.13. *A class of functions $\mathcal{F}$ is graph uniform Δ^0_2 if and only if it is upwards uniform Δ^0_2.*

Proof. Suppose that $\mathcal{F}$ is graph uniform Δ^0_2, and let $\{G_{e,s}\}_{e,s\in\omega}$ be a uniform Δ^0_2 approximation of the class of graphs of $\mathcal{F}$. Define the computable ternary function f as follows. For all $e, x \in \omega$, $f(e, x, 0) = 0$, and for any $s \in \omega$,

$$f(e,x,s+1) = \begin{cases} \mu y < s[G(e, \langle x, y\rangle, s+1) = 1] & \text{if } x < s \text{ and,} \\ & \text{there exists such } y, \\ s & \text{otherwise.} \end{cases}$$

It is straightforward to check that f is indeed an upwards uniform Δ^0_2 approximating function and that $\mathcal{F} = \{f_e\}_{e\in\omega}$ with upwards uniform Δ^0_2 approximation $\{f_{e,s}\}_{e,s\in\omega}$.

Now suppose that f is an upwards uniform Δ^0_2 approximating function for $\mathcal{F}$. Define the computable ternary function $G(e,s,x)$ as follows. For all $e,x,y,s \in \omega$,

$$G(e,\langle x,y\rangle,s+1) = \begin{cases} 1 & \text{if } f_{e,s}(x) = y, \\ 0 & \text{otherwise.} \end{cases}$$

Again it is easy to check that $\{G_{e,s}\}_{e,s\in\omega}$ is a uniform Δ^0_2 approximation and that $\{G_e\}_{e\in\omega}$ is precisely the class of graphs of $\mathcal{F}$. □

Lemma 3.14. *For any uniform Δ^0_2 class $\mathcal{A} \subseteq P(\omega)$, the class $\mathcal{F}_{\mathcal{A}} =_{def} \{f : G(f) \in \mathcal{A}\}$ is graph uniform Δ^0_2.*

Proof. Suppose that $\{A_{e,s}\}_{e,s\in\omega}$ is a uniform Δ^0_2 approximation of the class $\mathcal{A}$. Define the ternary computable function f as follows.

$$f(e,s+1,x) = \begin{cases} \mu y < s[A(e,\langle x,y\rangle,s+1) = 1] & \text{if } x < s \text{ and,} \\ & \text{there exists such } y, \\ s & \text{otherwise.} \end{cases}$$

Similarly to the first part of the proof of **Lemma 3.13**, it is straightforward to check that f is indeed an upwards uniform Δ^0_2 approximating function and that $\mathcal{F}_{\mathcal{A}} = \{f_e\}_{e\in\omega}$ with upwards uniform Δ^0_2 approximation $\{f_{e,s}\}_{e,s\in\omega}$. Thus, by **Lemma 3.13**, $\mathcal{F}_{\mathcal{A}}$ is graph uniform Δ^0_2. □

Lemma 3.15 (Ershov)**.** *For any $a \in \mathcal{O}$, Σ^{-1}_a is a uniform Δ^0_2 class.*

Proof. Note firstly that $\Sigma^{-1}_a = \emptyset$ for $a \in \{1,2\}$ (i.e. $|a|_{\mathcal{O}} \in \{0,1\}$) and $\Sigma^{-1}_a = \Sigma^0_1$ if $a = 4$ (i.e. $|a|_{\mathcal{O}} = 2$) which is clearly uniform Δ^0_2 with uniform Δ^0_2

approximation $\{W_{e,s}\}_{e,s\in\omega}$. Hence, we suppose that $a >_{\mathcal{O}} 4$, and we note by the previous sentence that $\Sigma^0_1 \subseteq \Sigma^{-1}_a$ in this case.

We suppose that $\{(f_e, o_e)\}_{e\in\omega}$ is a computable listing of all pairs of (respectively) $\{0,1\}$ and ω valued partial computable functions defined on $\omega \times \omega$ with associated uniform c.e. approximations $\{f_{e,s}\}_{e,s\in\omega}$ and $\{o_{e,s}\}_{e,s\in\omega}$.

We define a uniform Δ^0_2 approximation $\{A_{e,s}\}_{e,s\in\omega}$ such that $\Sigma^{-1}_a = \{A_e\}_{e\in\omega}$. The construction of the latter uses the following parameters. $l(e,s) \in \omega$ is the *level*, $w(e,s) \in \omega$ is the *witness* and satisfies $0 \leq w(e,s) \leq l(e,s)$, whereas $S(e,s) \in \{\mathsf{continue}, \mathsf{stop}\}$ is the *state*.

<u>*Stage* $s = 0$.</u> Set $l(e,0) = w(e,0) = A(e,n,0) = 0$ and $S(e,0) = \mathsf{continue}$ for all $e, n \in \omega$.

<u>*Stage* $s+1$.</u> For all $e > s$, reset $l(e,s+1) = w(e,s+1) = 0$ and $S(e,s+1) = \mathsf{continue}$ and reset $A(e,n,s+1) = 0$ for all $n \in \omega$.

For each $e \leq s$, process e according to which of the two cases below holds.

<u>*Case* 1.</u> $S(e,s) = \mathsf{stop}$. Then reset $S(e,s+1) = \mathsf{stop}$ and $A(e,n,s+1) = A(e,n,s)$ for all $n \in \omega$. (Also reset $u(e,s+1) = u(e,s)$ for $u \in \{l,w\}$. However, both of these parameters are now redundant.)

<u>*Case* 2.</u> $S(e,s) = \mathsf{continue}$. Let $n = l(e,s)$ and $m = w(e,s)$ (for clarity) and note that $0 \leq m \leq n \leq s$. Proceed as follows.

- For all $x \in \omega - \{m\}$, reset $A(e,x,s+1) = A(e,x,s)$.
- Let $r = n - m$ and test (1)-(5) below in order, stopping at the first test that fails. (The reader is referred back to **Theorem 1.30** for the definition of the functions p and q.)

(1) $f_{e,s+1}(m,r)\downarrow$.

(2) $o_{e,s+1}(m,r)\downarrow$.

(3) $o_{e,s+1}(m,r) \in W_{p(a),s+1}$.

(4) $r > 0$ & $o_{e,s+1}(m,r) \neq o_{e,s+1}(m,r-1) \Rightarrow$

$$\langle o_{e,s+1}(m,r), o_{e,s+1}(m,r-1)\rangle \in W_{q(a),s+1}.$$

(5) $r > 0$ & $f_{e,s+1}(m,r) \neq f_{e,s+1}(m,r-1) \Rightarrow$

$$o_{e,s+1}(m,r) \neq o_{e,s+1}(m,r-1).$$

<u>*Subcase* 2.*A*.</u> Test (i) fails for some $1 \leq i \leq 4$. Then reset $S(e,s+1) =$ continue, $l(e,s+1) = n$, $w(e,s+1) = m$ and $A(e,m,s+1) = A(e,m,s)$.

<u>*Subcase* 2.*B*.</u> Test (i) succeeds for all $1 \leq i \leq 4$ but fails for $i = 5$ (so definitively witnessing that the pair (f_e, o_e) does not define an a-c.e. set). In this case, set $S(e,s+1) =$ stop and $A(e,m,s+1) = A(e,m,s)$. (Also reset $u(e,s+1) = u(e,s)$ for $u \in \{l,w\}$. However, both of these parameters have now become redundant.)

<u>*Subcase* 2.*C*.</u> All the tests (i) for $1 \leq i \leq 5$ succeed. Then set $A(e,m,s+1) = f_{e,s+1}(m,r)$.

(a) If $m < n$, set $w(e,s+1) = m+1$ and $l(e,s+1) = n$.

(b) If $m = n$, set $w(e,s+1) = 0$ and $l(e,s+1) = n+1$.

Proceed to stage $s+2$.

This completes the description of the approximating function $A(e,n,s)$. It is now straightforward to check the following.

(i) $\{A_{e,s}\}_{e,s\in\omega}$ is a uniform Δ^0_2 approximation (i.e. that $\lim_{s\to\infty} A_{e,s}(n)$ exists for all $e, n \in \omega$).

(ii) For any set $A \subseteq \omega$, if $A \in \Sigma_a^{-1}$, then $A = A_e$ for some index e where the pair $(f_e, 0_e)$ witnesses this fact.

(iii) For every index e, either $A_e \in \Sigma_a^{-1}$ due to the fact that (f_e, o_e) witnesses this, or A_e is finite so that $A_e \in \Sigma_1^0 \subseteq \Sigma_a^{-1}$.

Therefore, $\{A_{e,s}\}_{e,s\in\omega}$ witnesses that Σ_a^{-1} is a uniform Δ_2^0 class. □

Corollary 3.16. *For any Σ_2^0 set $\mathcal{A} \subseteq \mathcal{O}$, $\Sigma_{\mathcal{A}}^{-1}$ is uniform Δ_2^0.*

Proof. Without loss of generality, we suppose that $\mathcal{A}$ contains some $a >_{\mathcal{O}} 4$. We adapt the proof of **Lemma 3.15** as follows. Let $\{\mathcal{A}_s\}_{s\in\omega}$ be a Σ_2^0 approximation to $\mathcal{A}$. Also, for any $a \in \omega$, let $A_{e,s}^a$ denote the stage s approximation defined relative to a by the construction in the proof of **Lemma 3.15**. Note that inspection of the latter shows that $A^a(e, n, s)$ is indeed well defined for all $e, n, s \in \omega$ even when $a \notin \mathcal{O}$. Likewise, if $a \in \mathcal{O}$, we will use $\{A_e^a\}_{e\in\omega}$ to denote the resulting uniform Δ_2^0 class.

We define a uniform Δ_2^0 approximation $\{A_{\langle a,e\rangle,s}\}_{a,e,s\in\omega}$. The construction use a *threshold* parameter $e(a, s)$ and a *relative stage* parameter $t(a, s)$.

Stage $s = 0$. Define $e(a, 0) = t(a, 0) = A(\langle a, e\rangle, n, 0) = 0$ for all $a, e, n \in \omega$.

Stage $s + 1$. For each $a > 0$ and for all $e, n \in \omega$, reset $e(a, s+1) = t(a, s+1) = 0$ and $A(\langle a, e\rangle, n, s + 1) = 0$.

For each $a \leq s$, we proceed according to the two cases below.

Case 1. $\mathcal{A}_s(a) = 0$ or $\mathcal{A}_{s+1}(a) = 0$. Then set $e(a, s + 1) = s + 1$, $t(a, s + 1) = 0$ and $A(\langle a, e\rangle, n, s + 1) = 0$ for all $e, n \in \omega$.

Case 2. Otherwise. (So $\mathcal{A}_s(a) = \mathcal{A}_{s+1}(a) = 1$.) Then reset $e(a, s+1) = e(a, s)$. Also set $t(a, s+1) = t(a, s) + 1$ and define

$$A(\langle a, e\rangle, n, s{+}1) = \begin{cases} A^a(e - e(a, s+1), n, t(a, s+1)) & \text{if } e \geq e(a, s+1), \\ 0 & \text{otherwise,} \end{cases}$$

for all $e, n \in \omega$.

Proceed to stage $s + 2$.

This completes the description of the approximating function $A(\langle a, e\rangle, n, s)$. It is now straightforward to check the following.

(i) $\{A_{\langle a,e\rangle,s}\}_{a,e,s\in\omega}$ is a uniform Δ^0_2 approximation.

(ii) If $a \notin \mathcal{A}$ then $A_{\langle a,e\rangle} = \emptyset$ for every index e.

(iii) If $a \in \mathcal{A}$, and e_a is the least stage such that $a \in \mathcal{A}_s$ for all $s \geq e_a$, then

  (a) $A_{\langle a,e\rangle} = \emptyset$ for all $e < e_a$.

  (b) $A_{\langle a,e\rangle} = A^a_{e-e_a}$ for all $e \geq e_a$.

Therefore, $\{A_{\langle a,e\rangle,s}\}_{a,e,s\in\omega}$ witnesses that $\Sigma^{-1}_{\mathcal{A}}$ is a uniform Δ^0_2 class. □

Remark 3.17. It follows from **Theorem 1.34** and **Corollary 3.16** that the set $\mathcal{A} = \{a : |a|_{\mathcal{O}} = \omega^2\} \subseteq \mathcal{O}$ is not Σ^0_2. Indeed, by **Theorem 1.34**, we know that

$$\Delta^0_2 = \Sigma^{-1}_{\mathcal{A}}.$$

Moreover, Δ^0_2 is itself *not* a uniform Δ^0_2 class. Thus, by **Corollary 3.16**, $\mathcal{A} \notin \Sigma^0_2$.

Note 3.18. Notice that by **Definition** 1.38 and **Definition** 1.39, for any set $\mathcal{A} \subseteq \mathcal{O}$, we can use the terminology of "$\Sigma^{-1}_{\mathcal{A}}$-rigidity". Indeed, we say that for a set $\mathcal{A} \subseteq \mathcal{O}$, a linear ordering $\mathscr{L}$ is $\Sigma^{-1}_{\mathcal{A}}$-rigid if there exists no nontrivial automorphism f of $\mathscr{L}$ such that f is a-c.e. for some $a \in \mathcal{A}$.

3.3 Uniform Δ^0_2-Rigidity of Computable Order Type $\mathbf{2} \cdot \boldsymbol{\eta}$

Lemma 3.19 (Upwards Search Lemma). *If $\mathscr{L} = \langle L, <_L \rangle$ is a computable linear ordering of order type $\mathbf{2} \cdot \boldsymbol{\eta}$, $L = \omega$, $p : L \to L$ is the associated pairing function*[2]*, and f is a nontrivial automorphism of $\mathscr{L}$, then the set*

$$K_f = \{a : a \in L \quad \& \quad p(a) > a \quad \& \quad f(a) > a \quad \& \quad f\big(p(a)\big) > a\} \tag{3.1}$$

is infinite.

Proof. Suppose that f is a nontrivial automorphism of $\mathscr{L}$. Given a (finite) set S, define $\mathcal{D}_f(S) = \{b : (\exists a \in S)(\exists n \in \omega)[b = f^{n+1}(a)]\}$, i.e. $\mathcal{D}_f(S)$ is the set of descendants of S under f. Note that if $f(a) \neq a$ then $\mathcal{D}_f(\{a\})$ is an infinite subordering of $\mathscr{L}$. Indeed, supposing that $a \ R \ f(a)$ for some $R \in \{<_L, >_L\}$, then $f^n(a) \ R \ f^{n+1}(a)$ for all $n \geq 0$. Moreover, not only does the same observation clearly apply to $b = p(a)$ but $\{a, b\} \cap \mathcal{D}_f(\{a\}) \cap \mathcal{D}_f(\{b\}) = \emptyset$.

Let a_0 be the least number a such that $f(a) \neq a$. Then $f(a_0) > a_0$ since if $f(a_0) < a_0$ then $f\big(f(a_0)\big) = f(a_0)$ by definition of a_0, which contradicts our assumption that f is a nontrivial automorphism. Let $b_0 = p(a_0)$. Suppose that

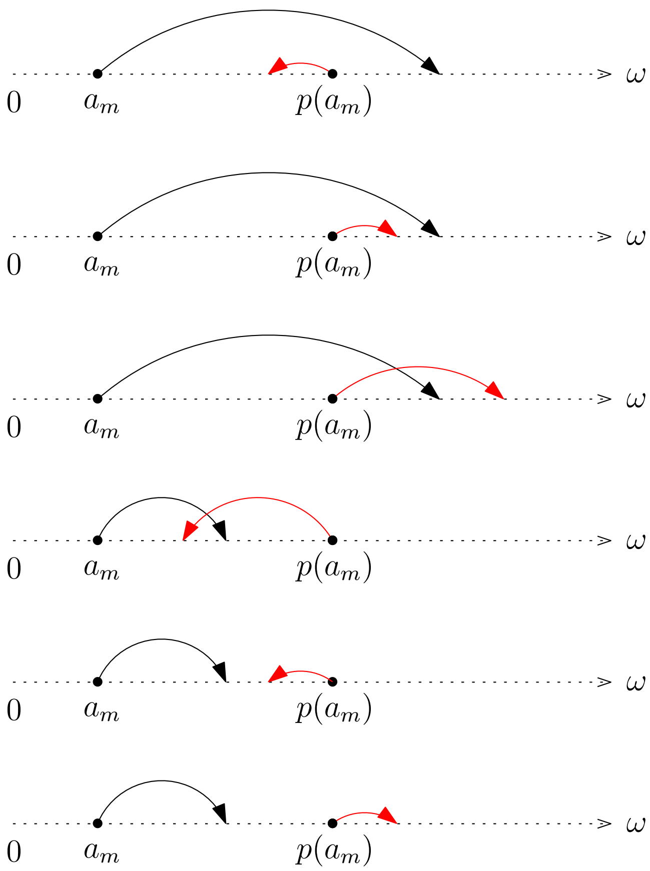
[2] p is a one-one and onto function with domain L such that for all $a \in L$, $p(a) \neq a$, but $p\big(p(a)\big) = a$ whereas $p(a)$ is either the $<_L$ predecessor or successor of a in $\mathscr{L}$ (i.e. no numbers lie $<_L$ between a and $p(a)$ in $\mathscr{L}$.)

$f(b_0) < a_0$. Then $f\big(f(b_0)\big) = f(b_0)$, again contradicting our assumption that f is a nontrivial automorphism. Thus $f(b_0) > a_0$. Therefore, either $f(b_0) > b_0$ and so $b_0 > a_0$ by definition of a_0, or otherwise $a_0 < f(b_0) < b_0$. So in both cases $a_0 < b_0$, whereas for each $c \in \{a_0, b_0\}$, $f(c) > a_0$.

Now suppose, as inductive hypothesis, that we have defined, for $n \geq 0$, the set $\{a_0, \ldots, a_n\}$ such that for all $m < n$, $a_m < a_{m+1}$ and

$$a_{m+1} = \mu a \in \mathcal{D}_f(\{a_m, p(a_m)\})[f(a) > a \ \& \ p(a) > a \ \& \ f\big(p(a)\big) > a]. \qquad (3.2)$$

Remark. In the set $\{a_0, \ldots, a_n\}$ *are six possible situations. Suppose that* $0 \leq m \leq n$. *The following arrows indicate mappings under* f.



Consider the pair $\{a_n, b_n\}$ where $b_n = p(a_n)$ (so that by assumption $b_n > a_n$.) Note firstly that we can deduce from the properties of f argued on above that

$$\mathcal{D}_f(\{a_n, b_n\}) \cap \{c : (\exists i < n)[c = a_i \vee c = p(a_i)]\} = \emptyset$$

(and that $\mathcal{D}_f(a_n) \cap \mathcal{D}_f(b_n) = \emptyset$.) Moreover,

$$|\{m : (\forall p < m)[f^{p+1}(c) < f^p(c)]\}| \leq c$$

for all each $c \in \{a_n, b_n\}$. In other words, there exists $b \in \mathcal{D}_f(\{a_n, b_n\})$ such that $f(b) > b$. Define a_{n+1} to be the least such b and set $b_{n+1} = p(a_{n+1})$. Suppose that $f(b_{n+1}) < a_{n+1}$. Then by definition of a_{n+1} (and our assumption that f is an automorphism) $f^{m+1}(b_{n+1}) < f^m(b_{n+1})$ for all $m \geq 0$, which is clearly a contradiction since $|\{n : n < b_{n+1}\}| = b_{n+1}$. Hence $f(b_{n+1}) > a_{n+1}$. So either $f(b_{n+1}) > b_{n+1}$ and hence $b_{n+1} > a_{n+1}$ by definition of a_{n+1}, or otherwise $a_{n+1} < f(b_{n+1}) < b_{n+1}$. So in both cases $a_{n+1} < b_{n+1}$, whereas $f(c) > a_{n+1}$ for each $c \in \{a_{n+1}, b_{n+1}\}$. Note also that, by definition of a_n, $a_{n+1} > a_n$. It follows that the set $\{a_0, \ldots, a_{n+1}\}$ satisfies the conditions of the inductive hypothesis, i.e. that the induction hypothesis is validated. We conclude therefore that K_f is indeed infinite. □

Theorem 3.20. *For any graph subuniform Δ^0_2 class $\mathcal{F}$, there exists a computable linear ordering $\mathscr{L}$ of order type $\mathbf{2} \cdot \boldsymbol{\eta}$ which is $\mathcal{F}$-rigid.*

Proof. We construct $\mathscr{L} = \langle L, <_L \rangle$ with associated pairing function p so that $L = \omega$. At each stage s, we define finite approximations to L, $<_L$ and p. L_s is defined to be an initial segment of ω such that $L_s \subset L_{s+1}$, and $<^s_L$ is defined with domain L_s. Note that by construction $<^s_L \subset <^{s+1}_L$ for all s. Accordingly during the construction we use the abbreviation $<_L$ instead of $<^s_L$. On the other hand, p_s is defined to be a one-one and onto function with domain L_s such that for all $a \in L_s$, $p_s(a) \neq a$, but $p_s\big(p_s(a)\big) = a$ whereas $p_s(a)$ is either the $<_L$ predecessor or successor of a in $\mathscr{L}_s$ (i.e. no numbers lie $<_L$ between a and $p_s(a)$ in $\mathscr{L}_s$.) The point here is that $p(n)$ is defined to be $\lim_{s\to\infty} p_s(n)$ if the latter exists — in which case we use the notation $p(n)\downarrow$ — and that we will require that $p(n)\downarrow$ for all $n \in \omega$ so that p is a total Δ^0_2 function. Note that we will also use the notation $P_s(a) = \{a, p_s(a)\}$ for any $a \in L_s$ and $P(a) = \{a, p(a)\}$ (under the supposition that $p(a)\downarrow$).

3.3.1 Requirements

Let $\mathcal{F}$ be a graph subuniform Δ^0_2 class of functions on ω. Accordingly, there exists a graph uniform Δ^0_2 class $\widehat{\mathcal{F}} = \{f_e\}_{e\in\omega}$ with upwards uniform Δ^0_2 approximation $\{f_{e,s}\}_{e,s\in\omega}$, such that $\mathcal{F} \subseteq \widehat{\mathcal{F}}$. The construction aims to satisfy for all $e \in \omega$, the following requirements

$$Q_e: \quad p(e)\downarrow,$$
$$R_e: \quad f_e \text{ is a nontrivial automorphism of } \mathscr{L};$$

the structural requirements

$$I: \quad (\forall n \in \omega)[p(n) \neq n \quad \& \quad p(p(n)) = n],$$
$$S: \quad (\forall n, m \in \omega)[m = p(n) \quad \Rightarrow \quad \{q : n <_L q <_L m\} = \emptyset],$$
$$T: \quad (\forall n, m \in \omega)[m \neq p(n) \;\; \& \;\; n <_L m \quad \Rightarrow \quad \exists q(m <_L q <_L n)];$$

and the complexity requirement

$$C: \quad \mathscr{L} \text{ is computable.}$$

Now the definition of p_s at each stage s will ensure that the requirements I and S (and the implicit requirement U that p is a well defined function) are satisfied — assuming that the requirements $\{Q_e\}_{e\in\omega}$ are satisfied — whereas the densification procedure in substage II of the construction (at stage s) will ensure that T is satisfied. On the other hand, the fact that $L_s \subset L_{s+1}$ and $<_L^s \subset <_L^{s+1}$ for all s ensures that requirement C is satisfied. Accordingly, the proof below is aimed at verifying that the sets of requirements $\{Q_e\}_{e\in\omega}$ and $\{R_e\}_{e\in\omega}$ are satisfied. It is then easily checked that satisfaction of these requirements entails that $\mathscr{L}$ is indeed a computable linear ordering of order type $\mathbf{2}\cdot\boldsymbol{\eta}$ which is $\mathcal{F}$-rigid. Firstly, we note that for a (total) function f to be an automorphism of $\mathscr{L}$ it must satisfy, for all $m, n \in \omega$ and $R \in \{<_L, >_L\}$, the following conditions.

$$\text{Order Preservation} \quad : \quad m\ R\ n \;\Leftrightarrow\; f(m)\ R\ f(n). \qquad \text{(OP)}$$

$$\text{Pair Preservation} \quad : \quad p\big(f(m)\big) \;=\; f\big(p(m)\big). \qquad \text{(PP)}$$

We also say that for any set $X \subseteq L$, a function $f : L \to L$ *commutes with* p *over* X if $f\big(p(a)\big) = p\big(f(a)\big)$ for all $a \in X$, and that f *preserves* $<_L$ *over* X if $a\ R\ b \Leftrightarrow f(a)\ R\ f(b)$ for all $a, b \in X$ and $R \in \{<_L, >_L\}$. (In other words, if f is an automorphism of $\mathscr{L}$, then f both commutes with p over X and preserves $<_L$ over X for any such X.)

3.3.2 Parameters for R_e

The construction works with a finite set of outcome constants

$$\mathcal{R} = \{\mathsf{wait}, \mathsf{dndiag}, \mathsf{updiag}, \mathsf{udiag}\} \tag{3.3}$$

with associated ordering $<_R$ so that $\mathsf{wait} <_R \mathsf{dndiag} <_R \mathsf{udiag} <_R \mathsf{updiag}$. The outcome parameter $r(e,s)$ is set to a value in $\mathcal{R}$. (Note that $r(e,s)$ is set to dndiag or a value in $\{\mathsf{updiag}, \mathsf{udiag}\}$ if the construction guesses at stage s that there is (respectively) a downwards or otherwise an upwards or partially downwards diagonalisation in place relative to R_e.) The parameter $L(e,s)$ contains a list of elements a such that the construction believes that $f_e(a) > a$, i.e. elements that are possibly eligible for attention. $a(e,s)$ denotes the greatest element in $L(e,s)$ if the latter is nonempty whereas, if $L(e,s) = \emptyset$, then $a(e,s)$ denotes the constant -1 (meaning *undefined*) ordered in the standard way relative to ω. $E(e,s)$ denotes the set $\bigcup_{a \in L(e,s)} P_s(a)$.

The overall outcome parameter $R(e,s)$ is set to the pair $\big(|L(e,s)|, r(e,s)\big)$. The parameters $F_0(e,s)$ and $F_1(e,s)$ are used to restrain pairs — and $|F_i| \in \{0,2\}$ for $i \in \{0,1\}$ — that the construction believes violate either (OP) or (PP). Accordingly, if $r(e,s) \in \{\mathsf{udiag}, \mathsf{updiag}\}$ then either $F_0(e,s) = P_s\big(f_{e,s}(a)\big)$ or $F_1(e,s) = P_s\big(f_{e,s}\big(p_s(a)\big)\big)$ where $a = a(e,s)$. If $r(e,s) \notin \{\mathsf{udiag}, \mathsf{updiag}\}$ on the other hand, then $F_0(e,s) = F_1(e,s) = \emptyset$. $F(e,s)$ denotes the set $F_0(e,s) \cup F_1(e,s)$ (so that $|F(e,s)| \in \{0,2,4\}$.)

The parameter $g(e,s)$ points to the maximum number in $E(e,s) \cup F(e,s)$ if one of the two sets is nonempty and otherwise to -1. The significance of $g(e,s)$ is that $\omega \restriction g(e,s)$ is restrained at stage s from injury by (i.e. re-pairing activity on behalf of) requirements R_i such that $i > e$. Accordingly, $g(e,s)$ indicates to the construction when processing such a requirement R_i the lowest threshold above which it can re-pair numbers without affecting previous (and still valid) action that it has taken on behalf of R_e. More precisely, an overall threshold $\hat{g}(i,s) = \max\{g(j,s) : j < i\}$ is used for R_i in the sense that any number n re-paired for the sake of R_i at stage s is such that $n > \hat{g}(i,s)$.

Remark 1. *We will show that for all indices* i, $\lim_{s\to\infty} g(i,s)$ *exists so that also, for any* e, $\lim_{s\to\infty} \hat{g}(e,s)$. *This allows the construction's action on behalf of* R_e *to work with the same lower threshold (of this type) at infinitely many stages.*

The parameter $t(e,s)$ points to the least number a such that the activity of R_e at stage s re-pairs a. If no such number exists, $t(e,s)$ points to $s-1$ (for $s>0$). $t(e,s)$ indicates to requirements R_i such that $i>e$ the threshold below which they can work. Accordingly, R_i only processes under an overall threshold $\hat{t}(i,s)$, where $\hat{t}(i,s)$ points to $\min\{t(j,s) : j<i\}$.

Remark 2. *We will show that for all* $e\in\omega$, $\lim_{s\to\infty} t(e,s)=\infty$. *Thus also, for all* $e\in\omega$, $\lim_{s\to\infty}\inf \hat{t}(e,s)=\infty$. *Hence, given index* e, *any number* n *lies under the threshold* $\hat{t}(e,s)$ *for requirement* R_e *for cofinitely many stages* s.

The parameter $c(e,a,s)\in\omega$ is used to defined a second type of lower threshold for re-pairing activity (via case A.8) carried out for the sake of requirement R_e. In detail, $c(e,a,s+1)$ points to $\min f_{e,s+1}[P_{e,s}(a)]$ when $a(e,s+1)=a$. From stage $s+1$, $c(e,a,s+1)$ is preserved up until (at least) stage $t>s+1$ provided that $a\in L(e,q)$ (i.e. $a(e,q)\geq a$) for all $s+1\leq q\leq t$. On the other hand, if a drops out of $L(e,t)$ at some stage $t>s+1$ (i.e. $a\in L(e,t-1)-L(e,t)$), then $c(e,a,t)$ is reinitialised to 0. The point here is that $c(e,a,t)$ indicates to the construction at stage $t+1$ a lower bound for a threshold below which it cannot re-pair numbers when applying case A.8 relative to some number $b=a(e,t+1)$ such that $b>a$. In particular, if $\lim_{s\to\infty}\inf a(e,s)=a$ and $f_e(d)\uparrow$ for each $d\in P(a)$ (and note that it is easily shown — via **Lemma**

3.21 — that $P(a) \downarrow$ in this case), then the construction runs the risk of re-pairing some number $c > a$ at infinitely many stages s when working on behalf of R_e via case A.8 relative to some (fixed) number $\hat{a} = a(e,s) > a$. (Note that this cannot happen for a itself since the fact that for each $d \in P(a)$, $f_e(d) \uparrow$ implied that $\lim_{s\to\infty} \inf f_{e,s}(d) = \infty$ by definition of $\widehat{\mathcal{F}}$.) The use by the construction of $c(e,a,s)$ removes this danger since the involvement of $c(e,a,s)$ as part of the threshold — $\hat{c}(e,\hat{a},s+1)$ defined below — for re-pairing activity undertaken relative to $\hat{a} = a(e,s+1)$ at stage $s+1$ means that eventually such activity will be permanently prohibited below $c+1$, since $\lim_{s\to\infty} \inf c(e,a,s) = \lim_{s\to\infty} \inf f_{e,s}[P(a)] = \infty$.

The parameter $\hat{c}(e,a,s+1)$ measures the maximum value of the set $\{c(e,b,s) : b < a\}$. The role of $\hat{c}(e,a,s+1)$, as indicated above, is to act as a lower threshold at or below which the construction cannot apply re-pairing activity for the sake of R_e relative to a at stage $s+1$ — i.e. when $a = a(e,s+1)$. More precisely, if $\min f_{e,s+1}[P_s(a)] \leq \hat{c}(e,a,s+1)$ then the construction cannot re-pair (via case A.8) $f_{e,s+1}[P_s(a)]$ at stage $s+1$.

Remark 3. *With the definition of* $\hat{c}(e,a,s)$ *in mind, the reader should note that the reason for the reinitialisation of* $\hat{c}(e,a,s)$ *to* 0 *if* a *is removed from* $L(e,s)$ *at stage* s*, is to prevent the scenario in which, for some* b *such that* $\min f_e[P(b)]$ *exists, a finite amount of eventually redundant activity occurring at numbers below* b *causes* b *to become ineligible for processing because this activity forces* $\hat{c}(e,b,t) > \min f_e[P(a)]$ *at cofinitely many stages* s*.*

3.3.3 Informal Overview of the Construction

For $i \in \omega$, let $\mathcal{M}_i$ denote the module of the overall construction which is dedicated to the satisfaction of requirement R_i. We consider the activity of $\mathcal{M}_e$ for given index e. We note firstly that $\mathcal{M}_e$ works under two overall assumptions. The first — assumption (A) — is that $\liminf_{s\to\infty} \hat{g}(e,s)\downarrow$ and the second — assumption (B) — is that $\liminf_{s\to\infty} \hat{t}(e,s) = \infty$. Assumption (A) implies that there is a minimum number $g = \liminf_{s\to\infty} \hat{g}(e,s) + 1$ at or above which, for infinitely many stages s, $\mathcal{M}_e$ can re-pair numbers without causing injury to the activity of higher priority modules — i.e. modules $\mathcal{M}_j$ such that $j < e$. Assumption (B) on the other hand implies that for any number n there exists a stage $s_{e,n}$ such that n is not re-paired by higher priority modules at any stage $s \geq s_{e,n}$. The action of $\mathcal{M}_e$ at stage $s+1$ will ensure that the set $L(e, s+1)$ is empty or else consists of a list of numbers a such that $\mathcal{M}_e$ sees that $f_{e,s+1}(a) > a$. Note that this in effect means that $\mathcal{M}_e$ guesses at stage $s+1$ that either $f_e(a)\downarrow > a$ or $f_e(a)\uparrow$ (so that, by definition $f_e \in \widehat{\mathcal{F}}$, in this case $\liminf_{s\to\infty} f_{e,s}(a) = \infty$ if $\mathcal{M}_e$'s guess is indeed correct). If $a \in L(e, s+1)$ then $P_{s+1}(a) = P_s(a)$ and, letting $b = p_s(a)$, if $\mathcal{M}_e$ sees that $f_{e,s+1}(b) > b$ then it will necessarily be the case that $b > a$. Also if $L(e, s+1) \neq \emptyset$, then the maximum number contained by $L(e, s+1)$ — i.e. $a(e, s+1)$ — is such that, for all numbers c such that $c < a(e, s+1)$ or such that c is at present paired — i.e. $d = p_s(c)$ — with some number $d \in L(e, s+1)$ such that $d < a(e, s+1)$, $\mathcal{M}_e$ has seen at stage $s+1$ that $f_{e,s+1}(c) = f_{e,s}(c)$. Moreover the definition of the restraint bound $\hat{g}(i, s+1)$ implies that for all $d \in L(e, s+1)$, $P_{s+1}(d) = P_s(d)$ due to the fact that $E(e, s+1) = \bigcup_{a \in L(e,s+1)} P_s(a) \subseteq \{0, \ldots, g(e, s+1)\}$ and that the initial segment $\{0, \ldots, g(e, s+1)\}$ is restrained by $\mathcal{M}_e$ from re-pairing activity by lower priority modules $\mathcal{M}_i$ via the definition of $\hat{g}(i, s+1)$ during (the ensuing part of) stage $s+1$.

On the other hand, over and above the role of $a(e,s)$ in measuring an initial segment over which f_e appears to converge (and over which the pairing function p will also converge by $\mathcal{M}_e$'s use of the restraint bound $g(e,s)$) the parameter $a(e,s)$ also acts as a focus of the activity of $\mathcal{M}_e$. Now $\mathcal{M}_e$ works under the assumption (C) that $\liminf_{s\to\infty} a(e,s)\downarrow$ and — letting $\hat{a}$ designate this assumed value — that $\lim_{s\to\infty} p_s(\hat{a})\downarrow$ if $\hat{a} > -1$. Then at any stage $s+1$ $\mathcal{M}_e$ also acts under the assumption (D_{s+1}) that, not only is $a(e,s+1) = \hat{a}$, but that $a(e,t) \geq a(e,s+1)$ and $p_t(\hat{a}) = p_s(\hat{a})$ — i.e. $= p(\hat{a})$ — for all $t \geq s+1$. Accordingly to assumption (C), there exists an infinite set T of stages $s+1$ such that the assumption (D_s) is correct at stage $s+1$. Let $\hat{b}$ designate the assumed value of $p(\hat{a})$ if $\hat{a} > -1$. We now describe the outcome of $\mathcal{M}_e$'s activity under these assumptions over the set of stages T. We do this by looking at the different reasons for which the situation $\liminf_{s\to\infty} a(e,s)\downarrow = \hat{a}$ (caused by the activity of $\mathcal{M}_e$) can arise.

Remark 4. *The reader should bear in mind that $\mathcal{M}_e$ only re-pairs a number n at stage $s+1$ if $n \in \{f_{e,s+1}(a), f_{e,s+1}(b) = J\}$ (say), where $a = a(e,s+1)$ and $b = p_s(a)$, and $n > \max\{a, \hat{g}(e,s+1), \hat{c}(e,a,s+1)\}$ (see case* A.8*). Note also that by definition of $a \in L(e,s)$, $b > a$ in this case. (This is because — noting firstly that $f_{e,s+1}(d) = f_{e,s}(d)$ for $d \in \{a,b\}$ since otherwise case A.8 would not apply — either $f_{e,s}(b) > b$ so $a < b$ since otherwise $b \in L(e,s)$ and $a \notin L(e,s)$, or otherwise $a < f_{e,s+1}(b) < b$.)*

(1) $\hat{a} > -1$ *and* $f_e(\hat{a})\downarrow$ *and* $f_e(\hat{b})\downarrow$. *Thus by definition of* $f_e \in \widehat{F}$,

$$\liminf_{s\to\infty} \min f_{e,s}[P(\hat{a})] = \infty.$$

Now note that by definition of T, given any stage $s+1 \in T$, if $\mathcal{M}_e$ carries out re-pairing via case A.8 *below at some stage $t+1 > s+1$, such that in fact $a(e,t+1) = c > \hat{a}$ then this can only happen if the numbers involved are greater than the parameter $c(e,\hat{a},t)$. However the conditions described here imply that $\liminf_{s\to\infty} c(e,\hat{a},t) = \infty$ (as explained in the comments above about this parameter). This implies that $\liminf_{s\to\infty} t(e,s) = \infty$ — since re-pairing relative to $\hat{a}$ also tends to infinity as at any stage $s+1$ this can only happen over the pair of numbers $f_{e,s+1}[P(\hat{a})]$. Moreover, at infinitely many stages $s+1 \in T$, $F(e,s+1) = F_0(e,s+1) \cup F_1(e,s+1) = \emptyset$ so that $\liminf_{s\to\infty} g(e,s)\downarrow$. On the other hand, this case implies that f_e is not total and hence not an automorphism of $\mathscr{L}$.*

(2) *$\hat{a} > -1$ $f_e(d)\downarrow$ and $f_e\big(p(d)\big)\uparrow$ for some $d \in P(\hat{a})$. Without loss of generality suppose that $d = \hat{b}$. Then in this case the activity of $\mathcal{M}_e$ may at some stage $s+1 \in T$ cause $f_e(\hat{b})$ to be*[3] *definitively re-paired with some number m (and note that this means that $f_e(\hat{b}) > \hat{a}$ in this case), by setting the restraint $F_1(e,t+1) = \{f_e(\hat{b},m)\}$ for all $t \geq s$. In this case by definition (see case* A.8*) the condition* (OP) *is violated from stage $s+1$ onwards. This outcome will not come about only if there exists a stage r' such that for all $r+1 \geq r'+1$, $f_{e,r+1}(\hat{a}) \neq p_r\big(f_e(\hat{b})\big)$. One case in which this happens is when either $f_e(\hat{b}) < \hat{a}$ or $f_e(\hat{b}) < \hat{g} = \liminf_{s\to\infty} \hat{g}(e,s)$. However in this case there exists some stage $s+1$ such that $P_{t+1}\big(f_e(\hat{b})\big) = P_{s+1}\big(f_e(\hat{b})\big)$ for all $t \geq s$ due to the restraint conditions attached to the parameters $a(e,s)$ and $\hat{g}(e,s)$ (and so in fact $\mathcal{M}_e$ sets $F_1(e,t+1) = P_{s+1}\big(f_e(\hat{b})\big)$ for all such t.) On the other hand, if $f_e(\hat{b}) > \hat{a}$ and $f_e(\hat{b}) > \hat{g}$ then there will in any case be a stage $t+1$ at which $F_1(e,s+1)$ is set permanently to $P_t\big(f_{e,t+1}(\hat{b})\big)$ for all $s \geq t$ since $\mathcal{M}_e$ sees that* (PP) *is violated — as $f_{e,t+1}(\hat{a}) \neq p_t\big(f_{e,t+1}(\hat{b})\big)$.*

[3]Note that we are assuming that $f_{e,s+1}(\hat{b})$ has already converged to $f_e(\hat{b})$.

Note that we are ensured of having one of these three different outcomes when reason (2) *is valid due to the fact that* $\liminf_{s\to\infty}(\hat{a}) = \infty$ *— in particular in the case when* $f_e(\hat{b}) < \hat{a}$ *or* $f_e(\hat{b}) < \hat{g}$ *— since this condition implies that* $f_{e,s+1}(\hat{a}) = p\big(f_e(\hat{b})\big)$ *for only finitely many stages* s*. Also note that in each of these cases there exists a stage* s^* *such that for all* $t \geq s^*$*,* $a(e,t) = \hat{a}$ *and* $F_i(e,t) = F_i(e,s^*)$ *for each* $i \in \{0,1\}$*. This means also that* $\lim_{s\to\infty} g(e,s) \downarrow= g(e,s^*)$ *and that* $\liminf_{s\to\infty} t(e,s) = \infty$ *(since* $\mathcal{M}_e$ *does not carry out any further re-pairing activity once* $F_0(e,s)$ *and* $F_1(e,s)$ *are permanently fixed.) Moreover in each case* R_e *is clearly satisfied.*

(3) $\hat{a} > -1$ *and both* $f_e(\hat{a})\downarrow$ *and* $f_e(\hat{b})\downarrow$ *or* $\hat{a} = -1$*. There are three possible outcomes in this case. Suppose that* $\hat{s}$ *is a stage such that* $a(e,t) \geq \hat{a}$ *for all* $t \geq \hat{s}$*.*

(i) $\hat{a} > -1$ *and there exists a stage* $s^* \geq \hat{s}$ *such that* $\mathcal{M}_e$ *permanently fixes either* $F_0(e,s) = P_{s^*}\big(f_e(\hat{a})\big)$ *or* $F_1(e,s) = P_{s^*}\big(f_e(\hat{b})\big)$ *for all stages* $s \geq s^*$*. This situation corresponds to two of the cases described in* (2)*, either due to re-pairing (case* A.8*) relative to* $\hat{a}$ *causing* (OP) *to be violated over* $\{\hat{a}, \hat{b}, f_e(\hat{a}), f_e(\hat{b})\}$ *at stage* s^* *or simply because* $\mathcal{M}_e$ *sees that either* (OP) *or* (PP) *is violated over* $\{\hat{a}, \hat{b}, f_e(\hat{a}), f_e(\hat{b})\}$ *at stage* s^**. In this case* $\lim_{s\to\infty} a(e,s)\downarrow= a(e,s^*)$*,* $\lim_{s\to\infty} g(e,s)\downarrow= g(e,s^*)$ *whereas also* $\liminf_{s\to\infty} t(e,s) = \infty$ *for the same reasons as those given in* (2)*. Also* R_e *is clearly satisfied.*

(ii) $\hat{a} > -1$ *and there exists a stage* $t^* \geq \hat{s}$ *such that* $\mathcal{M}_e$ *case* A.9 *permanently applies relative to* $\hat{a}$ *from stage* t^* *onwards. In other words, there is a set* $H \subseteq \omega \upharpoonright \hat{a}$ *such that* $H = c, d, f_e(c), f_e(d)$ *violates either* (OP) *or* (PP)*, and this can be seen by* $\mathcal{M}_e$ *at any stage* $s \geq t^*$ *because by definition of* $\hat{a}$ *and* t^**,* $f_{e,s}(c) = f_e(c)$ *and* $f_{e,s}(d) = f_e(d)$ *for all* *stages $s \geq t^*$. In this case $a(e,s) = \hat{a}$ and $F(e,s{+}1) = \emptyset$ for all $s \geq t^*$. In other words, $\lim_{s\to\infty} a(e,s) = a(e,t^*)$ and $\lim_{s\to\infty} g(e,s) = g(e,t^*)$. Also at no stage $s \geq s^*$ does $\mathcal{M}_e$ undertake any re-pairing activity, so that $\liminf_{s\to\infty} t(e,s) = \infty$. Moreover, R_e is clearly satisfied.*

(iii) *$\hat{a} \geq -1$ and neither (i) nor (ii) applies. This means that for any $n > \hat{a}$, $\mathcal{M}_e$ eventually sees that $f_e(n)$ converges to some number $m \leq n$. But this implies that K_{f_e} is finite so that f_e is not a nontrivial automorphism of $\mathscr{L}$ by* **Lemma 3.19**. *Note that in this case, there exists $\hat{r} \in T$ such that $F(e,s) = \emptyset$ for all $s \in \{t : t \geq \hat{r}\ \&\ t \in T\}$ so that not only is it the case that $E(e,s) = E(e,\hat{r})$, but also $g(e,s) = g(e,\hat{r})$ at all such stage, i.e. $\liminf_{s\to\infty} g(e,s){\downarrow} = g(e,\hat{r})$. Consider any $m \in \omega$. Then either $m \in E(e,\hat{r})$ so that by definition of $\hat{r}$, m cannot be re-paired by $\mathcal{M}_e$ at any stage $s \geq \hat{r}$ (since this would imply either that $a(e,s) = \hat{a}$ and $F(e,s) \neq \emptyset$, or that $a(e,s) < \hat{a}$, in contradiction with the definition of $\hat{r}$) or otherwise $m \notin E(e,\hat{r})$, in which case there is a stage $\hat{t} \geq \hat{r}$ such that for all $g(e,\hat{r}) < m' < m$, $f_e(m')$ has already converged to some $p' \leq m'$. But then, by Remark 4, m cannot be re-paired by $\mathcal{M}_e$ at any stage $s \geq \hat{t}$. It follows from this that $\liminf_{s\to\infty} t(e,s) = \infty$.*

Remark. *We can also of course reason directly without the use of* **Lemma 3.19** *in case (iii). Accordingly, suppose that $\hat{a} = -1$. Thus either for all $n \in \omega$, $f_e(n) = n$ so that f_e is the identity automorphism, or otherwise there exists some n such that $f_e(n) < n$. Let $\hat{n}$ be the least such n. Let $\hat{m} = f_e(\hat{n})$. Then $f_e(\hat{m}) = \hat{m}$ by definition of $\hat{n}$. Hence, f_e is not an automorphism in this case since for some $R \in \{<_L, >_L\}$, $\hat{n}\ R\ \hat{m}$ but it is not the case that $f_e(\hat{n})\ R\ f_e(\hat{m})$ since $f_e(\hat{n}) = f_e(\hat{m})$*

(i.e. (OP) *is violated.) A similar argument applies if* $\hat{a} > -1$.

We now consider the validity of assumptions (A), (B) and (C). We show firstly that under assumptions (A) an (B), assumption (C) is valid. Indeed, suppose that $\liminf_{s\to\infty} a(e,s) = \infty$. Define $I_e = \{a : f_e(a)\uparrow\}$ and suppose that $I_e \neq \emptyset$. Then it is easy to see that under assumption (B) (i.e. that $\liminf_{s\to\infty} \hat{t}(e,s) = \infty$) $\liminf_{s\to\infty} a(e,s) \downarrow\leq \min I_e$. Hence $\liminf_{s\to\infty} a(e,s) = \infty$ (and so correspondingly $\liminf_{s\to\infty} |L(e,s)| = \infty$) then $f_e(b)\downarrow$ for all $b \in \omega$. Moreover, by the argument found at the end of the first paragraph of this informal overview, we see that it is also the case that $\lim_{s\to\infty} p_s(b)\downarrow$ (i.e. $p_s(b)\downarrow$) for all $b \in \omega$. Now, if K_{f_e}, as defined in (3.1), is infinite then, as $\liminf_{s\to\infty} \hat{g}(e,s)$ exists by assumption (A), $\mathcal{M}_e$ will either (I) at some stage s^*+1 permanently re-pair a pair $P_{s^*}\big(f_e(a)\big)$ for some $a \in K_{f_e}$ causing $a(e,s) = a(e,s^*+1) = a$ for all $s \geq s^*$ or (II) discover that f_e either violates (OP) or (PP) over $\{a, b, f_e(a), f_e(b)\}$ where $a = p(b)$ or (III) discover that there exists some set $H \subseteq \omega \restriction a$ such that H violates one of these two conditions (see case A.9 below). However, this implies that $\liminf_{s\to\infty} a(e,s)\downarrow\leq a$. Likewise if K_{f_e} is finite, $\liminf_{s\to\infty} a(e,s)\downarrow$ for similar reasons. Thus, under assumptions (A) and (B), assumption (C) is valid. Now notice that we saw in cases (1)-(3) that $\liminf_{s\to\infty} g(e,s)\downarrow$ and that $\liminf_{s\to\infty} t(e,s) = \infty$. However, this implies that if assumption (A) and (B) hold for e (i.e. relative to $\mathcal{M}_e$), then they also hold for $e+1$ and are thus validated by induction over $e \in \omega$. Moreover, as for any number n, only the module $\mathcal{M}_e$ such that $e < n$ can re-pair n proves that n is only re-paired finitely often. It follows therefore that $\lim_{s\to\infty} p_s(n)\downarrow$ for all $n \in \omega$.

3.3.4 Construction

At stage 0, $L_0 = \emptyset$, $L(e,0) = E(e,0) = F_0(e,0) = F_1(e,0) = \emptyset$, $r(e,0) = \mathsf{wait}$, $a(e,0) = g(e,0) = \hat{g}(e,0) = -1$, $t(e,0) = \hat{t}(e,0) = 0$ and $c(e,a,0) = \hat{c}(e,a,0) = 0$ for all indices e and numbers $a \in \omega$.

At each stage $s+1$ the construction defines a finite initial segment of ω to be the domain L_{s+1} of the stage $s+1$ approximation $\mathscr{L}_{s+1}$ to $\mathscr{L}$, such that $\omega \restriction s+1 \subseteq L_{s+1}$.

Stage $s+1$.

There are two substages — I and II — at stage $s+1$. Substage I is dedicated to satisfying R_e for $e < s$ whereas substage II is dedicated to densification of $\mathscr{L}$, i.e. the satisfaction of requirement S.

Substage I.

This involves s steps. At step $e < s$ the construction processes requirement R_e. Each step involves two parts which we denote as parts A and B. We describe below step e (so that all $i < e$, R_i has already been processed at this stage.)

Notation. During the description of stage $s+1$ we use the notation f_e and f_e^- as shorthand for $f_{e,s+1}$ and $f_{e,s}$ respectively.

Step e: Part A. Begin by setting

$$\hat{t}(e,s+1) = \min\{t : t = s+1 \vee (\exists i < e)[t(i,s+1) = t]\} \tag{3.4}$$

and

$$\hat{g}(e, s+1) = \max\{g(i, s+1) : i < e\} \tag{3.5}$$

and for all $a < \hat{t}(e, s+1)$,

$$\hat{c}(e, a, s+1) = \max\{c(e, b, s) : b < a\}. \tag{3.6}$$

(Note here that due to reinitialisation, if $b \notin L(e, s)$ then $c(e, b, s) = 0$.)

The construction searches for the least $a \leq \hat{t}(e, s+1)$ such that one of the cases A1-A10 holds. If more than one case applies to this number a it chooses the first case in this list. It then processes the chosen case.

Remark 5. *One approach here might be, on the strength of* **Lemma 3.19**, *to only search at stage* $s+1$ *— and as possible candidates of* $L(e, s+1)$ *— for numbers* a *such that* $p_s(a) > a$ *and both* $f_e(a) > a$ *and* $f_e\big(p_e(a)\big) > a$. *However, this approach has the defect of being reliant to write into the strategy an inductively provable method of ensuring that the pairing function* p *is* Δ^0_2. *Accordingly, the approach taken here is to broaden the search to any number* a *such that it appears that* $f_e(a) > a$ *(i.e. in the limit), and keeping such numbers in* $L(e, s+1)$ *(so that* $P_s(a)$ *is restrained in* $\mathscr{L}_s$.) *The point here is that at any stage* t *the numbers contained in the set* $\bigcup_{a \in L(e,t)} P_t(a)$ *form a subordering of* $\mathscr{L}_t$, *and if* $L(e, t)$ *gradually grows (i.e. if it appears that* $\liminf_{s\to\infty} a(e, s) = \infty$*) then at some stage* s *onwards either the construction verifies that one of* (OP) *or* (PP) *is permanently violated below some* $a \in L(e, s)$ *(in which case* $\lim_{s\to\infty} a(e, s)$ *exists and points to the least such* a*), or otherwise that — perhaps due to the re-pairing activity carried out at* R_e *—* (PP) *is violated over the set* $\{a, b, f_e(a), f_e(b)\}$ *where* $a = a(e, s)$ *and* $b = p_s(a)$.

Notation. In cases A.1-A.8 below, b is used to denote $p_s(a)$ — so that $P_s(a) = \{a, b\}$.

<u>*Case* A.1.</u> $a < \hat{t}(e, s+1)$, $f_e(a) \neq f_e^-(a)$ and $f_e(a) \leq a$.

(Note that if $a \in L(e, s)$, then $f_e^-(a) > a$ by definition of $L(e, s)$. Also note that this case only happens finitely often for any given a by definition of $\mathcal{F}$.)

Then set $L(e, s+1) = L(e, s) \upharpoonright a$ (so that $a \notin L(e, s+1)$), $F_0(e, s+1) = F_1(e, s+1) = \emptyset$, and $r(e, s+1) = \mathsf{wait}$.

<u>*Case* A.2.</u> $a < \hat{t}(e, s+1)$, $f_e(a) \neq f_e^-(a)$ and $f_e(a) > a$.

Proceed according to the following subcases.

<u>*Case* A.2.i</u> $a \in L(e, s)$.

Then set $L(e, s+1) = L(e, s) \upharpoonright a \cup \{a\}$. Define $F_0(e, s+1) = \emptyset$ and

$$F_1(e, s+1) = \begin{cases} \emptyset & \text{if } f_e(b) \neq f_e^-(b) \text{ or} \\ & \text{if } F_1(e, s) \neq \emptyset \\ & \qquad \& \ \hat{t}(e, s+1) \leq \min F_1(e, s) \text{ or} \\ & \text{if } a \neq a(e, s), \\ F_1(e, s) & \text{otherwise.} \end{cases} \tag{3.7}$$

Note that if $F_1(e, s+1) \neq \emptyset$ then it must be the case that $a = a(e, s)$ and $r(e, s) \in \{\mathsf{updiag}, \mathsf{udiag}\}$. Also define

$$r(e, s+1) = \begin{cases} \mathsf{udiag} & \text{if } F_1(e, s+1) \neq \emptyset, \\ \mathsf{wait} & \text{otherwise.} \end{cases} \tag{3.8}$$

<u>*Case* A.2.ii</u> $b \in L(e,s)$ and $b < a$.

(Hence $f_e(b) = f_e^-(b)$ in this case.)

Then set $L(e,s+1) = L(e,s) \upharpoonright b \cup \{b\}$. Define $F_1(e,s+1) = \emptyset$ and

$$F_0(e,s+1) = \begin{cases} \emptyset & \text{if } F_0(e,s) \neq \emptyset \\ & \quad \& \ \hat{t}(e,s+1) \leq \min F_0(e,s) \text{ or} \\ & \text{if } b \neq a(e,s), \\ F_0(e,s) & \text{otherwise.} \end{cases} \tag{3.9}$$

Note that if $F_0(e,s+1) \neq \emptyset$ then it must be the case that $b = a(e,s)$ and $r(e,s) \in \{\mathsf{updiag}, \mathsf{udiag}\}$. Also define

$$r(e,s+1) = \begin{cases} \mathsf{udiag} & \text{if } F_0(e,s+1) \neq \emptyset, \\ \mathsf{wait} & \text{otherwise.} \end{cases} \tag{3.10}$$

<u>*Case* A.2.iii</u> Otherwise.

(So either $b \in L(e,s)$ and $a < b$ so that $f_e^-(a) \leq a$ by definition of $L(e,s)$, or otherwise $a \notin E(e,s) =_{\text{def}} \bigcup_{c \in L(e,s)} P_s(c)$.)

Then set $L(e,s+1) = L(e,s) \upharpoonright a \cup \{a\}$, $F_0(e,s+1) = F_1(e,s+1) = \emptyset$, and $r(e,s+1) = \mathsf{wait}$.

<u>*Case* A.3.</u> $a < \hat{t}(e,s+1)$, $a \in L(e,s)$, and $f_e(b) \neq f_e^-(b)$.

(So $a < b$ if this is the first case to apply, and thus also $f_e(a) \neq f_e^-(a)$.)

Then set $L(e,s+1) = L(e,s) \upharpoonright a \cup \{a\}$. Define $F_1(e,s+1) = \emptyset$ and

$$F_0(e,s+1) = \begin{cases} \emptyset & \text{if } F_0(e,s) \neq \emptyset \\ & \quad \& \ \hat{t}(e,s+1) \leq \min F_0(e,s) \text{ or} \\ & \text{if } a \neq a(e,s), \\ F_0(e,s) & \text{otherwise.} \end{cases} \tag{3.11}$$

Note that if $F_0(e, s+1) \neq \emptyset$ then it must be the case that $a = a(e,s)$ and $r(e,s) \in \{\mathsf{updiag}, \mathsf{udiag}\}$. Also define

$$r(e,s+1) = \begin{cases} \mathsf{udiag} & \text{if } F_0(e,s+1) \neq \emptyset, \\ \mathsf{wait} & \text{otherwise.} \end{cases} \tag{3.12}$$

<u>*Case* A.4.</u> $a < \hat{t}(e,s+1)$, $a \notin E(e,s)$, $f_e(a) \neq f_e^-(a)$ and $f_e(a) > a$.

Then set $L(e,s+1) = L(e,s) \restriction a \cup \{a\}$, $F_0(e,s+1) = F_1(e,s+1) = \emptyset$, and $r(e,s+1) = \mathsf{wait}$.

<u>*Case* A.5.</u> $a < \hat{t}(e,s+1)$, $a = a(e,s)$, $r(e,s) \in \{\mathsf{updiag}, \mathsf{udiag}\}$, and for some $i \in \{0,1\}$ such that $F_i(e,s) \neq \emptyset$, $\hat{t}(e,s+1) \leq \min F_i(e,s+1)$.

Set $L(e,s+1) = L(e,s)$ (so that $a(e,s+1) = a(e,s)$) and for $i \in \{0,1\}$, define $F_i(e,s+1) = \emptyset$ if $F_i(e,s) \neq \emptyset$ and $\hat{t}(e,s+1) \leq \min F_i(e,s)$. Otherwise set $F_i(e,s+1) = F_i(e,s)$.

$$r(e,s+1) = \begin{cases} \mathsf{updiag} & \text{if, for some } i \in \{0,1\},\ F_i(e,s+1) \neq \emptyset, \\ \mathsf{wait} & \text{otherwise.} \end{cases} \tag{3.13}$$

Case A.6. $a < \hat{t}(e, s+1)$, $a \in L(e, s)$, $\max\{b, f_e(a), f_e(b)\} < \hat{t}(e, s+1)$, and either $p_s\big(f_e(a)\big) \neq f_e(b)$ or for some $R \in \{<_L, >_L\}$, $a \; R \; b$ whereas it is not the case that $f_e(a) \; R \; f_e(b)$.

(Note that if this is the first case to apply then $f_e(a) > a$. Also note that this includes the case $f_e(a) = f_e(b)$.)

There are two subcases.

Case A.6.i $a = a(e, s)$ and $r(e, s) \in \{\mathsf{udiag}, \mathsf{updiag}\}$.

Set $L(e, s+1) = L(e, s)$, $F_i(e, s+1) = F_i(e, s)$ for $i \in \{0, 1\}$, and $r(e, s+1) = r(e, s)$.

Case A.6.ii Otherwise.

Set $L(e, s+1) = L(e, s) \restriction a \cup \{a\}$. Define $F_0(e, s+1) = P_s\big(f_e(a)\big)$, $F_1(e, s+1) = P_s\big(f_e(b)\big)$, and $r(e, s+1) = \mathsf{updiag}$.

Case A.7. $a < \hat{t}(e, s+1)$, $b < \hat{t}(e, s+1)$, for some $c \in \{a, b\}$, $f_e(c) < \hat{t}(e, s+1)$ and either $p_s\big(f_e(a)\big) \neq f_e(b)$ or for some $R \in \{<_L, >_L\}$, $a \; R \; b$ whereas it is not the case that $f_e(a) \; R \; f_e(b)$.

Set $L(e, s+1) = L(e, s) \restriction a \cup \{a\}$ and $r(e, s+1) = \mathsf{udiag}$. Define $F_0(e, s+1) = P_s\big(f_e(c)\big)$ and $F_1(e, s+1) = \emptyset$ if $c = a$. Otherwise — i.e. when $c = b$ — define $F_1(e, s+1) = P_s\big(f_e(c)\big)$ and $F_0(e, s+1) = \emptyset$.

Case A.8. $e < a < \hat{t}(e, s+1)$, $a \in L(e, s)$, and[4] $f_e(b) > a$, whereas also f_e preserves $<_L$ and commutes with p_s over[5] $P_s(a)$ — so that $P_s\big(f_e(a)\big) =$

[4]Note that if this is the first case to apply, then also $a < b$ and $a < f_e(a)$ by definition.

[5]Note that if this case is chosen by the construction, then case A.6 does not apply, so these two conditions follow by definition.

$\{f_e(a), f_e(b)\}$ — and

$$\max\{\hat{g}(e, s+1), \hat{c}(e, a, s+1)\} < \min P_s\big(f_e(a)\big) < \hat{t}(e, s+1). \tag{3.14}$$

In this case choose the least numbers n, m not yet enumerated into L. Supposing that $f_e(a)\ R\ f_e(b)$ (for some $R \in \{<_L, >_L\}$), and n, m to $\mathscr{L}$ such that $p_{s+1}\big(f_e(a)\big) = n$, $p_{s+1}(n) = f_e(a)$, $p_{s+1}\big(f_e(b)\big) = m$, $p_{s+1}(m) = f_e(b)$, and $n\ R\ f_e(a)$ whereas $f_e(b)\ R\ m$ (and define m, n appropriately under $<_L$ relative to all other numbers in L at this point in the construction.)

Set $L(e, s+1) = L(e, s) \upharpoonright a \cup \{a\}$, $F_0(e, s+1) = P_{s+1}\big(f_e(a)\big)$, $F_1(e, s+1) = P_{s+1}\big(f_e(b)\big)$, and $r(e, s+1) = \mathsf{updiag}$.

<u>*Case* A.9.</u> $a < \hat{t}(e, s+1)$, $a \in L(e, s)$ and there exists numbers c, d such that the set $H = \{c, d, f_e(c), f_e(d)\}$ satisfies $a > \max H$ and either case (a) or case (b) below applies.

(a) H violates (OP) in the sense that for $R \in \{<_L, >_L\}$, $c\ R\ d$ but it is not the case that $f_e(c)\ R\ f_e(d)$.

(b) H violates (PP) in the sense that for $(S, S') \in \{(=, \neq), (\neq, =)\}$, $d\ S\ p_s(c)$ whereas $f_e(d)\ S'\ p_s\big(f_e(c)\big)$.

Then set $L(e, s+1) = L(e, s) \upharpoonright a \cup \{a\}$, $F_0(e, s+1) = F_1(e, s+1) = \emptyset$ and $r(e, s+1) = \mathsf{dndiag}$.

<u>*Case* A.10.</u> $a = \hat{t}(e, s+1)$.

Set $L(e, s+1) = L(e, s) \restriction a$, $F_0(e, s+1) = F_1(e, s+1) = \emptyset$, and $r(e, s+1) = \textsf{wait}$.

<u>*Step e*: *Part* B.</u> Set $R(e, s+1) = (|L(e, s+1)|, r(e, s+1))$. Set

$$a(e, s+1) = \begin{cases} \max L(e, s+1) & \text{if } L(e, s+1) \neq \emptyset, \\ -1 & \text{otherwise.} \end{cases} \tag{3.15}$$

Define

$$E(e, s+1) = \bigcup_{a \in L(e,s+1)} P_s(a) \tag{3.16}$$

and note that for all $a \in L(e, s+1)$, $P_s(a) = P_{s+1}(a)$ by construction. Also define

$$F(e, s+1) = \bigcup_{i \in \{0,1\}} F_i(e, s+1) \tag{3.17}$$

and

$$g(e, s+1) = \max E(e, s+1) \cup F(e, s+1) \cup \{-1\} \tag{3.18}$$

and if $a = a(e, s+1) \neq -1$, then set

$$c(e, a, s+1) = \min f_e[P_s(a)] \tag{3.19}$$

($= \min\{f_e(a), f_e\big(p_s(a)\big)\} = \min P_{s+1}\big(f_e(a)\big) \cup P_{s+1}\big(f_e(p_s(a))\big)$ by construction — where of course these two latter pairs may be identical.) For all $b > a(e, s+1)$ set $c(e, b, s+1) = 0$ and note that for all $d < a(e, s+1)$, by automatic resetting $c(e, d, s+1) = c(e, d, s)$. Set

$$t(e, s+1) = \begin{cases} \min P_s\big(f_e(a)\big) & \text{if case A.8 applies,} \\ s & \text{otherwise.} \end{cases} \tag{3.20}$$

(Note that in (3.20) $\min P_s\big(f_e(a)\big) = \min F(e, s+1)$ by definition.) If $e < s$ go to step $e+1$ of substage I. Otherwise go to substage II.

Substage II. (Resetting and Densification)

At the end of substage I the construction has defined the finite linear ordering $\widehat{\mathscr{L}_{s+1}}$ such that $\widehat{\mathscr{L}_s}$ is a subordering of $\widehat{\mathscr{L}_{s+1}}$. Let $m = |\widehat{L}_{s+1}|$. Then $m = 2n$ for some n. Accordingly suppose that $\widehat{\mathscr{L}_{s+1}} = \{b_0 <_L b_1 <_L \ldots <_L b_{2n-2} <_L b_{2n-1}\}$. Then firstly, for all $b \in \widehat{L}_{s+1}$ such that $p_s(b)$ was not redefined during substage I, set $p_{s+1}(b) = p_s(b)$ (so that $P_{s+1}(b) = P_s(b)$.) Secondly, letting $c_0, \ldots, c_{2n+1}$ be the next $2(n+1)$ numbers in $\omega - \widehat{L}_{s+1}$. Define $\mathscr{L}_{s+1}$ by setting $L_{s+1} = \widehat{L}_{s+1} \cup \{c_0, \ldots, c_{2n+1}\}$, defining

$$c_{2i} <_L c_{2i+1} <_L b_{2i} <_L b_{2i+1} <_L c_{2i+2} <_L c_{2i+3}$$

for all $i < n$, and setting $p_{s+1}(c_{2j}) = c_{2j+1}$ and $p_{s+1}(c_{2j+1}) = c_{2j}$ for all $j \leq n$.

Proceed to stage $s+2$.

3.3.5 Verification

For clarity we consider the construction from the point of view of a tree of outcomes as follows. We firstly set $\Lambda = \omega \times \mathcal{R}$ where $\mathcal{R}$ is the set of outcome constants defined in (3.3) with associated ordering $<_R$. We suppose that Λ has an associated lexicographical ordering $<_{\text{lex}}$, so that for any $(n, r), (m, \hat{r}) \in \Lambda$, $(n, r) <_{\text{lex}} (m, \hat{r})$ if either $n < m$ or otherwise $n = m$ and $r <_R \hat{r}$. Now the reader will notice that at stage s of the construction, we define a path $\beta \in \Lambda^{<\omega}$

(of overall outcomes) of length s defined by setting $\beta(i) = R(i, s)$ for all $i < s$. Accordingly, we use the notation α_s to designate the path of length s defined at stage s. Also for all $\beta \in \Lambda^{<\omega}$ we say that stage s is β-*true* if $\beta \subseteq \alpha_s$. We will show that for every $e \in \omega$ there exists a least $\beta \in \Lambda^{<\omega}$ (under $<_{\text{lex}}$) of length e such that the set $\{s : s \text{ is } \beta\text{-true}\}$ is infinite. We will use δ_e to designate this string and δ to designate the infinite path $\gamma \in \Lambda^\omega$ defined by setting $\gamma \upharpoonright e = \delta_e$ for all e. The significance of δ can be seen by considering the activity of the construction on behalf of any requirement R_e. Indeed, let s be a δ_e-true stage such that $\alpha_t \not<_{\text{lex}} \delta_e$ for all $t \geq s$. Then by construction, we will be able to show that this implies that at every stage $r \geq s$, $\hat{g}(e, r) \leq \hat{g}(e, t)$ and moreover that at every subsequent δ_e-true stage t, $\hat{g}(e, s) = \hat{g}(e, t)$. In other words, activity on behalf of R_e works with a fixed finite restraint at infinitely many (δ_e-true) stages.

The following Lemma can be checked by a straightforward inspection of the construction.

Lemma 3.21. *Let s be any stage. Then the following conditions hold.*

(a) *For all $b \leq a(e, s+1)$, $p_{s+1}(b) = p_s(b)$.*

(b) *For all $b < a(e, s+1)$, $f_{e,s+1}(b) = f_{e,s}(b)$.*

(c) *For all b such that $p_s(b) \in L(e, s+1) - \{a(e, s+1)\}$, $f_{e,s+1}(b) = f_{e,s}(b)$.*

(d) $a(e, s) \leq g(e, s)$.

(e) *For all b such that $b \leq g(e, s+1)$, $p_{s+1}(b) = p_s(b)$ except in the case when $b \in P_s\Big(f_{e,s+1}\big(a(e, s+1)\big)\Big)$ and case 8 holds at stage $s+1$.*

(f) *For each $(P, Q) \in \{(\subseteq, \leq), (=, =)\}$, $L(e, s)\ P\ L(e, s+1)$ iff $a(e, s)\ Q\ a(e, s+1)$.*

(g) *If* $a \in L(e,s)$ *and* $f_{e,s}\big(p_s(a)\big) > p_s(a)$, *then* $p_s(a) > a$.

Notation 3. We use α_s to denote the empty string λ if $s = 0$ and otherwise the string

$$\Big(\big(|L(0,s)|, r(0,s)\big), \ldots, \big(|L(s-1,s)|, r(s-1,s)\big)\Big) \tag{3.21}$$

if $s > 0$. We say that α_s is the stage s path. For any string $\beta \in \Lambda^{<\omega}$ we say that stage s is β-true if $\beta \subseteq \alpha_s$.

Definition 3.22. Define $\delta_e = \mu\beta[|\beta| = e \ \&\ \forall s(\exists t > s)(\beta \subseteq \alpha_s)]$ where we define μ to be the function that finds the least string β under $<_{\text{lex}}$ satisfying the conditions in the following box $[\ldots]$, so that $\delta_e \uparrow$ if there exists no such β.

Lemma 3.23. *For all* $e \in \omega$, $\delta_e \downarrow$.

Proof. We proceed by induction on $e \in \omega$. Note that the case $e = 0$ is trivially true since $\lambda \subseteq \alpha_s$ for all stages s, i.e. $\delta_0 = \lambda$. We thus consider the case $e+1$ under the induction hypothesis that δ_e exists. Accordingly there are infinitely many stages s such that $\delta_e \subseteq \alpha_s$ and there exists a δ_e-true stage s_e such that for all $s \geq s_e$, $\alpha_s \not<_{\text{lex}} \delta_e$. As part of the induction hypothesis we will also assume that for every stage $s \geq s_e$, $\hat{g}(e, s_e) \leq \hat{g}(e,s)$ and that if s is δ_e-true, then $\hat{g}(e,s) = \hat{g}(e,s_e)$.

Suppose that $\delta_e \uparrow$. It follows that $\liminf_{s\to\infty} a(e,s+1) = \infty$ and that for all $b \in \omega$, $\forall t(\exists s > t)[b \in L(e,s)] \Rightarrow \exists t(\forall s > t)[b \in L(e,s)]$ so if we define $L(e) = \{a : \forall t(\exists s \geq t)[a \in L(e,s)]\}$ we see that $L(e)$ is an infinite Δ^0_2 set. Notice also that by definition of case A.10, $\liminf_{s\to\infty} \hat{t}(e,s) = \infty$ in this case. Now note that $\liminf_{s\to\infty} a(e,s+1) = \infty$ implies, by **Lemma 3.21**, that for all $b \in \omega$, $\lim_{s\to\infty} p_s(b)\downarrow$ and $\lim_{s\to\infty} f_{e,s}(b)\downarrow$. In particular, f_e is a total function. Now note firstly that f_e is not the identity automorphism in this case. Indeed, suppose that f_e is the identity automorphism and that b and s are such that

$b \in L(e,s)$. Then under the additional assumption that $a(e,r) \geq b$ for all $r \geq s$, we can see there will exist a stage $t \geq s$ such that case A.1 will apply to b at stage $t+1$. But then $a(e,t+1) < b$; a contradiction. Hence, we can deduce in this case that $\liminf_{s\to\infty} a(e,s) = -1$. Thus, either (a) f_e is nontrivial automorphism or (b) f_e violates either (OP) or (PP). Suppose that (b) is the case and that $H = \{c, d, f_e(c), f_e(d)\}$ witnesses this. Let a be the least number in $L(e) \cap \{n : n > \max H\}$. Let s^* be a stage such that $f_{e,s}(b) = f_e(b)$ for each $b \in \{c,d\}$ and $p_s(b') = p(b')$ for each $b' \in H$ for all $s \geq s^*$. Thus, case A.9 will apply at every such stage s. Hence, $\liminf_{s\to\infty} a(e,s) \leq a$. Contradiction.

Hence, f_e must be a nontrivial automorphism of $\mathscr{L}$. Define $\hat{g} = \hat{g}(e, s_e)$, and define $\widehat{L} = L(e) \upharpoonright \hat{g}+1$. Note that under our assumptions $\lim_{s\to\infty} c(e,a,s)$ exists for all $a \in \omega$. We use $c(e,a)$ to denote this value. Define $\hat{c} = \max\{c(e,a) : a \in \widehat{L}\}$. Let $\hat{t}$ be the least stage such that $a(e,s) > \hat{g}$ for all stages $s \geq \hat{t}$. By **Lemma 3.19** there exists a such that $p(a) > a$ and $\min f_e[P(a)] > \max\{a, \hat{g}, \hat{c}\}$. Let $\hat{a}$ be the least such a. Let $\hat{s}$ be the least stage $s \geq \hat{t}$ such that $a(e,s) > \hat{a}$ for all $s \geq \hat{s}$ (so that $\hat{a} \in L(e,s)$ by definition.) Then by definition, $p_{\hat{s}}(a) = p(a)$ and $f_{e,\hat{s}}(c) = f_e(c)$ for each $c \in \{a, p(a)\}$ and so case A.8 will apply at stage $\hat{s}$. In other words, $a(e,\hat{s}) \leq \hat{a}$. Contradiction.

We conclude therefore (see **Note 3.24** below) that $\liminf_{s\to\infty} a(e,s)$ exists and that therefore $\delta_{e+1}\downarrow$, since $\mathcal{R}$ is a finite set. Let t_{e+1} be a δ_{e+1}-true stage such that $\alpha_t \not<_{\text{lex}} \delta_{e+1}$ for all $t \geq t_{e+1}$. Then by **Lemma 3.21** it follows that $E(e, t_{e+1}) \subseteq E(e,s)$ for all $s \geq t_{e+1}$ and also that at every such stage s, if s is δ_{e+1}-true, then $E(e,s) = E(e,t_{e+1})$. It now remains to show that the same applies to $F(e,s)$. Accordingly, let

$$S = \{s : s \geq t_{e+1} \ \&\ \alpha_s \text{ is } \delta_{e+1}\text{-true}\}$$

and let $\hat{a} = \lim_{s \in S} a(e,s)$ and $\hat{r} = \lim_{s \in S} r(e,s)$. In other words, $\delta_{e+1} = (|\widehat{L}|, \hat{r})$ where $\widehat{L} = L(e, t_{e+1}) = L(e,s) \upharpoonright \hat{a}+1$ for all $s \geq t_{e+1}$.

Claim. *If $\hat{r} \in \{\mathsf{dndiag}, \mathsf{udiag}, \mathsf{updiag}\}$, then there exists a stage $s_{e+1} \geq t_{e+1}$ such that for all $s \geq s_{e+1}$, $b(e,s) = b(e, s_{e+1})$ for each $b \in \{a, r, F\}$.*

Proof. Note firstly that by definition of t_{e+1}, $a(e,s) \geq \hat{a}$ for all $s \geq t_{e+1}$.

Suppose that $\hat{r} = \mathsf{dndiag}$. Then, as $a(e,s) \geq \hat{(a)}$ for all $s \geq t_{e+1}$, we know that for all $b < \hat{a}$, $p_s(b) = p_{t_{e+1}}(b)$ and $f_{e,s}(b) = f_{e,t_{e+1}}(b)$ for all such s. It follows that the diagonalisation condition of case A.9 remains in place relative to $\hat{a}$ for all $s \geq t_{e+1}$. Hence, $a(e,s) = a(e,s+1)$, $r(e,s) = r(e,s+1)$ and $F(e,s) = \emptyset$ for all $s \geq t_{e+1}$.

Suppose that $\hat{r} = \mathsf{updiag}$. Then for all $s \geq t_{e+1}$ we can show by induction (on s) that by definition of $\hat{g}(j,s)$, no lower priority requirement R_j interferes with $F_0(e,s)$ and $F_1(e,s)$ and thus that by construction, $a(e,s) = a(e,s+1)$, $r(e,s) = r(e,s+1)$ and for each $i \in \{0,1\}$, $F_i(e,s) = F_i(e,s+1)$ for all such s (since otherwise $a(e,s+1) = \hat{a}$ and $r(e,s+1) < \mathsf{updiag}$ in contradiction with the definition of t_{e+1}.)

Suppose that $\hat{r} = \mathsf{udiag}$. Then similarly to the case $\hat{r} = \mathsf{updiag}$, $a(e,s) = \hat{a}$ for all $s \geq t_{e+1}$ whereas $r(e,s) \in \{\mathsf{udiag}, \mathsf{updiag}\}$ by definition of t_{e+1}. Suppose firstly that $r(e,s) = \hat{r}$ for all such stages s. Then it must be the case that $F_i(e,s) = F_i(e,s+1) \neq \emptyset$ for some $i \in \{0,1\}$ (whereas $F_{1-i}(e,s) = F_{1-i}(e,s+1) = \emptyset$) for all $s \geq t_{e+1}$ since otherwise $r(e,s+1) = \mathsf{wait}$ in contradiction with the definition of t_{e+1}. Otherwise, at some (least) stage $t > t_{e+1}$, for all $s < t$, $r(e,s) = \mathsf{udiag}$ (and $a(e,s) = \hat{a}$) but $r(e,t) = \mathsf{updiag}$ due to case A.8 being applied at stage t relative to $\hat{a}$. However, in this case, one of the restraints

$F_0(e,t)$ or $F_1(e,t)$ will be preserved at every stage $s \geq t$. This is because, by definition of case A.8, and letting $\hat{b} = p(\hat{a})$, $f_e(\hat{a})$ is re-paired at stage t with a number n (i.e. $F_0(e,t) = \{f_e(\hat{a}), n\}$) such that $\hat{a}\ X\ \hat{b}$ iff $f_e(\hat{a})\ \overline{X}\ n$ and $f_e(\hat{b})$ is re-paired with a number m (i.e. $F_1(e,t) = \{f_e(\hat{b}, m)\}$) such that $\hat{b}\ X\ \hat{a}$ iff $f_e(\hat{b})\ \overline{Y}\ b$ where $(X, \overline{X}), (Y, \overline{Y}) \in \{(<_L, >_L), (>_L, <_L)\}$. We can now show by induction on $s \geq t$ that for one (only) index $i \in \{0,1\}$, $F_i(e,s)$ is preserved. In particular, case A.8 can no longer apply since $F_i(e,s)$ witnesses that (OP) is violated at every such stage s. (And by definition of this case there will be a stage $q_{e+1} > t_{e+1}$ such that $r(e,s) \neq \mathsf{updiag}$ for all $s \geq a_{e+1}$ (and so $r(e,s) = \mathsf{udiag}$.)) □

We can now deduce from the **Claim** that if $\hat{r} \in \{\mathsf{dndiag}, \mathsf{udiag}, \mathsf{updiag}\}$, then there exists $\hat{t} \geq t_{e+1}$ such that $b(e,s) = b(e,\hat{t})$ for each $b \in \{a, r, F_0, F_1, g\}$ and all $s \geq \hat{t}$. On the other hand, if $\hat{r} = \mathsf{wait}$, then $g(e,s) = \max E(e,s)$ at all δ_{e+1}-true stages $s > t_{e+1}$. Hence, in both cases, there exists a stage $s_{e+1} \geq s_e$ such that $\hat{g}(e+1, s_{e+1}) \leq \hat{g}(e+1, s)$ for all $s \geq s_{e+1}$ and such that if α_s is δ_{e+1}-true, then $\hat{g}(e+1, s_{e+1}) = \hat{g}(e+1, s)$. Hence, the induction hypothesis is validated. This concludes the proof of **Lemma 3.23**. □

Notation 4. We use δ to denote the member of Λ^ω defined by setting $\delta \upharpoonright e = \delta_e$ for all e. We call δ the *true path*.

Notation 5. From now on, we use s_{e+1} to denote the least δ_{e+1}-true stage $\hat{s} > 0$ such that for all $t \geq \hat{s}$, $\alpha_t \not<_{\text{lex}} \delta_{e+1}$ and such that also, if $\liminf_{s\to\infty} r(e,s) \in \{\mathsf{dndiag}, \mathsf{udiag}, \mathsf{updiag}\}$, then for all such t, $b(e,t) = b(e,\hat{s})$ for each $b \in \{a, r, g\}$. We also define

$$S_e = \{s : s \geq s_{e+1}\ \&\ |L(e,s)| = |L(e,s_e)|\ \&\ r(e,s) = r(e,s_e)\},$$

(so that $S_e = \{s : s \geq s_{e+1}\}$ if $\liminf_{s\to\infty} r(e,s) \in \{\mathsf{dndiag}, \mathsf{udiag}, \mathsf{updiag}\}$.)

Note 3.24. Note that, implicit in the above arguments is the fact that for $s \geq s_{e+1}$, if $|L(e,s)| = |L(e,s_{e+1})|$, then $L(e,s) = L(e,s_{e+1})$ (since otherwise $\alpha_t <_{\text{lex}} \delta_{e+1}$ for some $t \geq s_{e+1}$) and thus also $a(e,s) = a(e,s_{e+1})$. Moreover, $r(e,s) = r(e,s_{e+1})$, $\hat{g}(e,s) = \hat{g}(e,s_{e+1})$ and $g(e,s) = g(e,s_{e+1})$. Accordingly, for $b \in \{L, a, r, \hat{g}, g\}$ we use $b(e)$ to denote $b(e,s_{e+1}) = \liminf_{s\to\infty} b(e,s)$. Notice that $\delta(e+1) = \big(|L(e)|, r(e)\big)$.

We assume as induction hypothesis in **Lemmas 3.25-3.26**, that $\lim_{s\to\infty} \hat{t}(e,s) = \infty$ and we note that the proof of **Lemma 3.26** validates the induction hypothesis.

Lemma 3.25. *If $a(e) \geq 0$, and for some $b \in P\big(a(e)\big)$, $\lim_{s\to\infty} f_e(b) \downarrow$ — whereas, letting $c = p(b)$, $\lim_{s\to\infty} f_{e,s}(c) \uparrow$ — then $r(e) \in \{\mathsf{dndiag}, \mathsf{udiag}, \mathsf{updiag}\}$. (So that $r(e) = \lim_{s\to\infty} r(e,s)$.)*

Proof. Let s_{e+1} be as in *Notation* 5 and b, c be as in the statement of **Lemma 3.25**. Suppose for a contradiction that $r(e) = \mathsf{wait}$. Let $u \geq s_e$ be a stage such that $f_{e,t}(b) = f_{e,u}(b)$, $\hat{t}(e,t) > \max\{\hat{g}(e), f_{e,u}(b)\}$, and $f_{e,t}(c) > f_{e,u}(b)$ for all $t \geq u$.

Suppose that $f_e(b) = \lim_{s\to\infty} f_{e,s}(b) \leq a(e)$ or that $f_e(b) \leq \hat{g}(e)$. Then it follows from **Lemma 3.21** that $P_t\big(f_e(b)\big) = P_u\big(f_e(b)\big)$ for all $t \geq u$ — so that $P\big(f_e(b)\big) = P_u\big(f_e(b)\big)$. Now let $v \geq u$ be a δ_{e+1}-true stage such that $f_{e,t}(c) > p\big(f_e(b)\big)$ for all $t \geq v$. Then case A.6 or case A.7 will apply at stage v relative to $a(e)$, so that $r(e,v) \in \{\mathsf{updiag}, \mathsf{udiag}\}$. However, this contradicts the assumption that v is a δ_{e+1}-true stage (which entails that $r(e,v) = \mathsf{wait}$.)

Hence, $f_e(b) > \max\{a(e), \hat{g}(e)\}$. Now notice that $d(e) = d(e,s_{e+1})$ for each $d \in \{a, \hat{g}\}$, i.e. $a(e)$ and $\hat{g}(e)$ have already entered L (the domain of $\mathscr{L}$) by stage

s_{e+1}, so for any $s \geq s_{e+1}$, any m entering L_s is such that $m > \max\{a(e), \hat{g}(e)\}$. Hence, if $p_t\big(f_e(b)\big) < \max\{a(e), \hat{g}(e)\}$ for some $t \geq s_{e+1}$, it must be the case that $p_r\big(f_e(b)\big) = p_{s_{e+1}}\big(f_e(b)\big)$ for all $s_{e+1} \leq r \leq t$. Thus, we have two cases to consider as follows. (i) $p_{s_{e+1}}\big(f_e(b)\big) \leq \max\{a(e), \hat{g}(e)\}$ or (ii) $p_{s_{e+1}}\big(f_e(b)\big) > \max\{a(e), \hat{g}(e)\}$.

If case (i) holds, then $p_{s_{e+1}}\big(f_e(b)\big)$ is permanently (i.e. from stage s_{e+1} onwards) restrained in $\mathscr{L}$ due to the constructions use of $a(e)$ and $\hat{g}(e)$. We thus get a contradiction just as in the earlier case of $f_e(b) \leq \max\{a(e), \hat{g}(e)\}$.

So consider case (ii). Then for all $s \geq u$ we know that $\min P_s\big(f_e(b)\big) > \max\{a(e), \hat{g}(e)\}$. Let $\hat{u}+1 > u+1$ be a δ_{e+1}-true stage (so that $a(e) = a(e, \hat{u}+1)$ and $\hat{g} = g(e, \hat{u}+1)$.) Suppose that $\min P_{\hat{u}}\big(f_e(b)\big) \leq \hat{c}(e, a(e), \hat{u}+1)$. Then by definition there exists some $a \in L(e, \hat{u}) \upharpoonright a(e)$ such that $\min f_{e,\hat{u}}[P_{\hat{u}-1}(a)] > \max\{a(e), g(e)\} > a$. In other words, $f_{e,\hat{u}}(a) > a$ and $f_{e,\hat{u}}\big(p_{\hat{u}-1}(a)\big) > a$. Also $p(a) = p_{\hat{u}-1}(a) = p_{\hat{u}}(a)$ since $a \in L(e) = L(e, \hat{u}-1) \upharpoonright a(e) \cup \{a(e)\}$ and $\hat{u} - 1 \geq s_{e+1}$. Thus $p_{\hat{u}}(a) > a$ by definition of $L(e)$. But then one of the cases A.6-8 applies at stage $\hat{u}+1$ so that $a_{\hat{u}+1} <_{\text{lex}} \delta_{e+1}$ in contradiction with the definition of s_{e+1}. Hence, $p_{\hat{u}}\big(f_e(b)\big) > \hat{c}(e, a(e), \hat{u}+1)$. However, this means that one of cases A.6-8 applies at stage $\hat{u}+1$ relative to $a(e)$. But this means that $r(e, \hat{u}+1) \in \{\mathsf{updiag}, \mathsf{udiag}\}$ whereas by assumption, $r(e, \hat{u}+1) = \mathsf{wait}$ since stage $\hat{u}+1$ is δ_{e+1}-true. Contradiction.

We conclude therefore that $r(e) \neq \mathsf{wait}$ so that $r(e) \in \{\mathsf{dndiag}, \mathsf{udiag}, \mathsf{updiag}\}$.

□

Lemma 3.26. *$\liminf_{s\to\infty} t(e,s) = \infty$.*

Proof. Let s_{e+1} be as in *Notation* 5. Suppose firstly that

$$r(e) \in \{\mathsf{dndiag}, \mathsf{udiag}, \mathsf{updiag}\}.$$

Note that this means that $a(e, s) = a(e)$ and $r(e, s) = r(e)$ for all $s \geq s_{e+1}$, case A.8 does not apply at any such stage $t > s_{e+1}$, although it might apply at stage s_{e+1} relative to $a(e)$ (see the proof of the **Claim**). Hence, $\hat{t}(e, s+1) = s$ for all $s \geq s_{e+1}$. I.e. $\lim_{s\to\infty} \hat{t}(e, s) = \liminf_{s\to\infty} \hat{t}(e, s) = \infty$ in this case.

So now suppose that $r(e) = \mathsf{wait}$. Then we deduce from **Lemma 3.25** that either

(i) $a(e) \geq 0$ and $\lim_{s\to\infty} f_{e,s}(b) \uparrow$ for each $b \in P\big(a(e)\big)$, or

(ii) $a(e) = -1$, or

(iii) $a(e) \geq$ and $\lim_{s\to\infty} f_{e,s}(b) \downarrow$ for each $b \in P\big(a(e)\big)$.

Consider case (i). Then since $\liminf_{s\to\infty} f_{e,s}(b) = \infty$ for each $b \in P\big(a(e)\big)$, the infimum over s of the re-pairing activity caused by case A.8 relative to $a(e)$ tends to ∞. Also this means that by definition, for any $b > a$, $\liminf_{s\to\infty} \hat{c}(e, b, s) = \infty$. Thus, for any such b there will be a stage s_b such that case A.8 never applies relative to b at stages $s \geq s_b$. It follows therefore that — since by definition, for any $a \in \omega$, case A.8 applied relative to a (for the sake of R_e) only re-pairs numbers that are greater than a — the infimum of *all* re-pairing activity carried out for the sake of requirement R_e tends to ∞. So we can deduce in case (i) that $\liminf_{s\to\infty} t(e, s) = \infty$.

Consider case (ii). Suppose that there exists $b \in \omega$ such that $\lim_{s\to\infty} f_{e,s} \uparrow$. Let a be the least such number. Then there exists a stage t_a such that $f_{e,s}(a) > a$ for all $s \geq t_a$ and so $a \in L(e, s)$ at every such stage s. However, this means that

$a(e) \geq a$. Contradiction. Therefore, $\lim_{s\to\infty} f_{e,s}(b) \downarrow$ for all $b \in \omega$. Moreover, it is not the case that $f_e(b) = \lim_{s\to\infty} f_{e,s}(b) > b$ since otherwise, by the argument used for a above now reapplied for b we can deduce that $a(e) \geq b$, again giving a contardiction. Hence, for every $a \in \omega$, case A.8 can only apply relative to a (for the sake of R_e) at finitely many stages. So again in case (ii) we deduce that $\liminf_{s\to\infty} t(e,s) = \infty$.

Consider case (iii). Then the argument of case (ii) apply to all $a \notin L(e)$ such that $a > a(e)$ to show that $\liminf_{s\to\infty} t(e,s) = \infty$ in this case also. □

Corollary 3.27. *For all $n \in \omega$, Q_n is satisfied.*

Proof. Consider any n. Notice firstly that n can only be re-paired by the activity of case A.8 carried out for the sake of some requirement R_e such that $e < n$. However, by definition, for any stage s, $t(e,s)$ is a lower bound for the re-pairing activity carried out for the sake of requirement R_e. Moreover, by **Lemma 3.26**, $\liminf_{s\to\infty} t(i,s) = \infty$ for all indices i. Thus, there exists a stage t_n such that $t(j,s) > n$ and $s \geq t_n$. But then $p_s(n) = p_{t_n}(n)$ for all such s. In other words, $\lim_{s\to\infty} p_s(n) \downarrow = p_{t_n}(n)$. □

Lemma 3.28. *For all e, R_e is satisfied.*

Proof. Suppose that f_e is a nontrivial automorphism of $\mathscr{L}$. Define $\hat{a}, \hat{g}$ and $\hat{c}$ as in the discussion of the case when f_e is a nontrivial automorphism in **Lemma 3.23**. Then by a similar argument we see that at some stage $\hat{s}$ case A.8 will be applied for the sake of R_e relative to $\hat{a}$ causing a permanent diagonalisation, i.e. contradicting the assumption that f_e is an automorphism. □

This concludes the proof of **Theorem 3.20**. □

We can now apply — with **Note 3.18** in mind — **Lemma 3.15**, **Corollary 3.16** and **Theorem 3.20** to the following.

Corollary 3.29. *For every Σ_2^0 set $\mathcal{A} \subseteq \mathcal{O}$, there exists a computable linear ordering of order type $\mathbf{2} \cdot \boldsymbol{\eta}$ which is $\Sigma_{\mathcal{A}}^{-1}$-rigid.*

3.4 Uniform Δ_2^0-Rigidity of Computable Order Type $\boldsymbol{\omega} + \boldsymbol{\zeta}$

Theorem 3.30. *For any graph subuniform Δ_2^0 class $\mathcal{F}$, there exists a computable linear ordering of order type $\boldsymbol{\omega} + \boldsymbol{\zeta}$ which is $\mathcal{F}$-rigid.*

Remark. *For any linear orderings $\mathscr{L}$, $\mathscr{A}$ and $\mathscr{B}$ such that $\mathscr{A}$ is of order type $\boldsymbol{\omega}$ and $\mathscr{B}$ is of order type $\boldsymbol{\zeta}$, and $\mathscr{L} = \mathscr{A} + \mathscr{B}$, and automorphism f of $\mathscr{L}$, $f(z) \neq z$ for all $z \in A$ (the domain of $\mathscr{A}$). Moreover if f is a nontrivial automorphism, then $f(z) \neq z$ for all $z \in B$ (the domain of $\mathscr{B}$).*

Proof. We construct $\mathscr{L} = \langle L, <_L \rangle$ so that $L = \omega$ and in such a way that $\mathscr{L} = \mathscr{B} + \mathscr{C}$ where $\mathscr{B} = \langle B, <_B \rangle$ is of order type $\boldsymbol{\omega}$, $\mathscr{C} = \langle C, <_C \rangle$ is of order type $\boldsymbol{\zeta}$ and $<_B$ and $<_C$ are the restrictions of $<_L$ to domains B and C respectively. Note firstly that as in **Theorem 3.20**, at each stage s, we define finite approximations to L and $<_L$. L_s is defined to be an initial segment of ω such that $L_s \subset L_{s+1}$ and $<_L^s$ is defined with domain L_s. Note that by construction $<_L^s \subseteq <_L^{s+1}$ for all s. (See **Lemma 3.31** below.) Accordingly, we use the abbreviation $<_L$ instead of $<_L^s$ during the construction. Now, in order for $\mathscr{L}$ to be of the right order type, we also define finite blocks $\mathscr{B}_s$ and $\mathscr{C}_s$ such that $\mathscr{L}_s = \langle L_s, <_L \rangle = \mathscr{B}_s + \mathscr{C}_s$. These blocks are defined so that $L_s = B_s \cup C_s$

where $\mathscr{B}_s = \langle B_s, <_{B_s}\rangle$ and $\mathscr{C}_s = \langle C_s, <_{C_s}\rangle$ and such that for $X \in \{B, C\}$, $<^s_{X_s}$ is simply the restriction of $<_L$ to domain X_s. (So again for simplicity we will only use the notation $<_L$ in place of $<^s_{X_s}$.) Moreover, these blocks are defined in such a way that by setting $\mathscr{X} = \liminf_{s\to\infty} \mathscr{X}_s$ for each $\mathscr{X} \in \{\mathscr{B}, \mathscr{C}\}$ — where this limit is taken under the subordering relation $\subset$ for linear orderings — it is indeed the case that $\mathscr{B}$ and $\mathscr{C}$ are of order type $\boldsymbol{\omega}$ and $\boldsymbol{\zeta}$ respectively.

3.4.1 Requirements

Let $\mathcal{F}$ be a graph subuniform Δ^0_2 class of functions on ω. Accordingly, there exists a graph uniform Δ^0_2 class $\widehat{\mathcal{F}} = \{f_e\}_{e\in\omega}$ with upwards uniform Δ^0_2 approximation $\{f_{e,s}\}_{e,s\in\omega}$, such that $\mathcal{F} \subseteq \widehat{\mathcal{F}}$. The construction aims to satisfy for all $e \in \omega$, the following requirements

$$R_e \quad : \quad f_e \text{ is not a nontrivial automorphism of } \mathscr{L};$$

the structural requirement

$$S \quad : \quad \mathscr{L} \text{ is of order type } \boldsymbol{\omega} + \boldsymbol{\zeta};$$

and the complexity requirement

$$C \quad : \quad \mathscr{L} \text{ is computable.}$$

Notation 6. During the construction we use $\langle\emptyset\rangle$ to denote the trivial linear ordering $\langle\emptyset, <_L\rangle$ and $\langle n\rangle$ to denote the singleton linear ordering $\langle\{n\}, <_L\rangle$ (for all $n \in \omega$).

3.4.2 Construction

The construction uses a witness parameter $x(e,s) \in \omega \cup \{\uparrow\}$ and two structural parameter $m(e,s)$, $p(e,s) \in \omega \cup \{\uparrow\}$.

Stage 0. Set $A_0 = \{0\}$, $B_0 = \{1\}$, so that $\mathscr{A}_0 = \langle 0 \rangle$ and $\mathscr{B}_0 = \langle 1 \rangle$. Set $\mathscr{L}_0 = \mathscr{A}_0 + \mathscr{B}_0$, ($L_0 = \{0,1\}$ and $\mathscr{L} = \langle L_0, <_L \rangle$ where $0 <_L 1$.) For all $e \in \omega$, $x(e,0) = m(e,0) = p(e,0) =\uparrow$.

Stage $s+1$. There are two substages to be processed.

Substage I. Let e be the least $i \leq s$ such that either $x(i,s) =\uparrow$ or otherwise $x(i,s) \in \omega$ and for $(\gamma, C) \in \{(=, A), (\neq, B)\}$, $f_{i,s}\big(x(i,s)\big)\ \gamma\ x(i,s)$ and $x(i,s) \in C_s$. (Note that one such case will always apply.)

Case A. $x(e,s) =\uparrow$. Then define $\widehat{\mathscr{A}_{s+1}} = \mathscr{A}_s$, $\widehat{\mathscr{B}}_{s+1} = \mathscr{B}_s$, $\mathscr{F}_{s+1} = \langle \emptyset \rangle$ and $\mathscr{G}_{s+1} = \langle \emptyset \rangle$. Proceed to substage II.

Case B. $(\gamma, C) = (=, A)$. Then set $\mathscr{F}_{s+1} = \langle F_{s+1}, <_L \rangle$ where

$$F_{s+1} =_{\text{def}} \{z : z = x(e,s)\ \vee\ [x(e,s) <_L z\ \&\ z \in A_s]\}$$

and set $\mathscr{G}_{s+1} = \langle \emptyset \rangle$. Define $\widehat{\mathscr{A}_{s+1}} = \langle A_s - F_{s+1}, <_L \rangle$ and $\widehat{\mathscr{B}}_{s+1} = \mathscr{B}$. Go to substage II.

Case C. $(\gamma, C) = (\neq, B)$. Then set $\mathscr{G}_{s+1} = \langle G_{s+1}, <_L \rangle$ where

$$G_{s+1} =_{\text{def}} \{z : z = x(e,s)\ \vee\ [z <_L x(e,s)\ \&\ z \in B_s]\}$$

and set $\mathscr{F}_{s+1} = \langle\emptyset\rangle$. Define $\widehat{\mathscr{B}}_{s+1} = \langle B_s - G_{s+1}, <_L\rangle$ and $\widehat{\mathscr{A}_{s+1}} = \mathscr{A}$. Go to substage II.

<u>Substage II.</u> Let m, n, p, q be the least numbers in $\omega - L_s$. Define

$$\mathscr{A} = \widehat{\mathscr{A}_{s+1}} + \langle m\rangle + \langle n\rangle + \mathscr{G}_{s+1}$$

and

$$\mathscr{B} = \mathscr{F}_{s+1} + \langle p\rangle + \widehat{\mathscr{B}}_{s+1} + \langle q\rangle$$

and define

$$\mathscr{L}_{s+1} = \mathscr{A}_{s+1} + \mathscr{B}_{s+1}$$

and notice that this means that $A_{s+1} = A_s \cup \{n, m\}$, $B_{s+1} = B_s \cup \{p, q\}$ and $L_{s+1} = A_{s+1} \cup B_{s+1} = L_s \cup \{m, n, p, q\}$. Now proceed according to whether either Case A or Case B-C applied.

(i) Case A applies (so that $x(e, s) =\uparrow$.) Then set $x(e, s+1) = n$, $m(e, s+1) = m$ and $p(e, s+1) = p$ and note that by definition of case A, $z <_L m <_L n$ for all $z \in A_s$ whereas $p <_L w$ for all $w \in B_s$. For all $i \neq e$ and $r \in \{x, m, p\}$ set $r(i, s+1) = r(i, s)$.

(ii) Case B or C applies. Then reset $x(e, s+1) = x(e, s)$, reinitialise all $k > e$ by setting $x(k, s+1) =\uparrow$, and set $x(i, s+1) = x(i, s)$ for all $i < e$. For $r \in \{m, p\}$ and all $j \in \omega$, reset $r(j, s+1) = r(j, s)$.

Finish the stage and go to stage $s + 2$.

3.4.3 Verification

We verify that the construction satisfied the requirements via the following Lemmas. (Note that **Lemma 3.31** justifies our use of the abbreviation $<_L$ instead of $<_L^s$ during the construction.)

Lemma 3.31. *For all stages s, $\mathscr{L}_s \prec \mathscr{L}_{s+1}$, in other words $L_s \subset L_{s+1}$ and $<_L^s \subset <_L^{s+1}$.*

Proof. This is obvious from inspection of the construction. □

Lemma 3.32. *For all $e \in \omega$ the following hold.*

(1) $\lim_{s\to\infty} x(e,s) \downarrow \in \omega$ *(and this value is denoted as $x(e)$.)*

(2) $\lim_{s\to\infty} m(e,s) \downarrow \in \omega$ *— denoted as $m(e)$ — and $\{z : 0 <_L z <_L m(e)\}$ is finite.*

(3) $\lim_{s\to\infty} p(e,s) \downarrow \in \omega$ *— denoted as $p(e)$ — and $\{z : p(e) <_L z <_L 1\}$ is finite.*

(4) $m(e) <_L m(e+1)$.

(5) $p(e+1) <_L p(e)$.

(6) *Requirement R_e requires attention at only finitely many stages.*

Proof. Consider some $e \in \omega$. We assume as inductive hypothesis that conditions (1)-(6) hold for all $i < e$. Accordingly, let s_e be the least stage such that for all $t > s_e$, $x(i,t) = x(i,s_e) \in \omega$ for all $i < e$ and R_i does not receive attention at any stage $t > s$. Inspection of the construction shows that $x(e, s_e + 1) \in \omega$ and moreover that $x(e,r) = x(e, s_e + 1)$ for all stages $r \geq s_e + 1$, since R_e can no longer be reinitialised. In other words $x(e) = x(e, s_e + 1)$. Likewise,

$m(e) = m(e, s_e{+}1)$ and $p(e) = p(e, s_e{+}1)$. Also it is clear from the construction that for all $j \geq e$ and $t \geq x_e + 1$, $m(e) <_L x(j,t) <_L p(e)$ by construction and that for all numbers $n \notin \omega - L_{s_e+1}$, n will be placed in the ordering $\mathscr{L}$ so that either $m(e) <_L n <_L p(e)$ or $1 <_L n$. It follows that each of the sets $\{z : 0 <_L z <_L m(e)\}$ and $\{z : p(e) <_L z <_L 1\}$ is finite.

Now, since $f_e \in \widehat{\mathcal{F}}$, we know that there exists a stage $t_e \geq s_e + 1$ such that for all stages $t \geq t_e$, either $f_{e,t}\big(x(e)\big) = x(e)$ or $f_{e,t}\big(x(e)\big) \neq x(e)$. It is clear therefore that R_e can receive attention at most once after stage t_e. Hence, R_e only receive attention finitely often. Now let $r_e \geq t_e$ be a stage such that R_e does not receive attention at any stage $s \geq r_e$. Then at stage r_e we will have that $m(e+1, r_e) \in \omega$ with $m(e) <_L m(e+1, r_e)$ and also that $p(e+1, r_e) \in \omega$ with $p(e+1, r_e) <_L p(e)$. By a similar argument to the one used above, $q(e+1, s) = q(e+1, r_e)$ for $q \in \{m, p\}$ and all $s \geq r_e$. In other words $m(e) <_L m(e{+}1) = m(e{+}1, r_e)$ whereas $p(e{+}1) = p(e{+}1, r_e) <_L p(e)$.

Lemma 3.32 is thus satisfied for e, under the assumption that the induction hypothesis holds. Hence the latter is validated and **Lemma 3.32** is proved. □

Lemma 3.33. *For all n, if $n <_L 1$ then there exists e such that either $0 <_L n <_L m(e)$ or otherwise $p(e) <_L n <_L 1$.*

Proof. Consider some $n \in \omega$. By construction there exists a stage s such that n enters L_s. Suppose that it is not the case that $1 <_L n$. Choose e such that $x(e, s) \uparrow$. Let $t+1 > s$ be the least stage such that $x(e, r) = x(e, t+1)$ for all $r \geq t+1$ — i.e. $x(e) = x(e, t+1)$. Notice that, as we saw in the proof of **Lemma 3.32**, this means that $q(e) = q(e, r)$ for $q \in \{m, p\}$ and all stages $r \geq t{+}1$. It now suffices to note that either $n \in A_t$ or $n \in B_t$ — where A_t and B_t are the domains of $\mathscr{A}_t$ and $\mathscr{B}_t$ respectively — and also that $z <_L m(e, t{+}1)$

for all $z \in A_t$ whereas $p(e, t+1) <_L z$ for all $z \in B_t$. (This can be seen from the fact that R_e receives attention at stage $t+1$ via substage II (i) thus ensuring that both $m(e, t+1)$ and $p(e, t+1)$ are set to numbers in $\omega - L_t$ in the manner described on page 89.) Hence, either $0 <_L n <_L m(e)$ or $p(e) <_L n <_L 1$. $\square$

Lemma 3.34. *$\mathscr{L}$ has order type $\boldsymbol{\omega} + \boldsymbol{\zeta}$.*

Proof. This lemma follows from **Lemma 3.32** (2)-(5), **Lemma 3.33**, and the obvious fact that $\mathscr{B}^*$ has order type $\boldsymbol{\omega}$, where the latter denotes $\mathscr{B}^* = \langle B^*, <_L \rangle$ with $B^* = \{n : n = 1 \ \vee \ 1 <_L n\}$. $\square$

Lemma 3.35. *For all $e \in \omega$, R_e is satisfied.*

Proof. Consider any $e \in \omega$. Let r_e be the stage defined in the proof of **Lemma 3.32**, i.e. so that R_e does not receive attention at any stage $t \geq r_e$. Then either, for all $t \geq r_e$, $f_{e,t}\big(x(e)\big) \neq x(e)$ and $x(e) \in A_t$ (the domain of $\mathscr{A}_t$) or $f_{e,t}\big(x(e)\big) = x(e)$ and $x(e) \in B_t$ (the domain of $\mathscr{B}_t$). Moreover, for all $i > e$ and stages $s \geq r_e$ such that $x(i, s) \in \omega$, if $x(e) in A_{r_e}$ then $x(e) <_L m(i, s) <_L x(i, s)$ whereas if $x(e) \in B_{r_e}$ then $x(i, s) <_L p(i, s) <_L x(e)$. Thus, for $C \in \{A, B\}$, if $x(e) \in C_{r_e}$ then $x(e) \in C$, where C is the domain of the corresponding linear ordering $\mathscr{C} \in \{\mathscr{A}, \mathscr{B}\}$.

Now suppose that f_e is a nontrivial automorphism of $\mathscr{L}$. Then in particular $f_e(n)\downarrow$ for all $n \in \omega$. Also it is easily seen that $f_e(m) = m$ for all $m \in A$ whereas $f_e(m) \neq m$ for all $m \in B$. This contradicts the fact that $f_e\big(x(e)\big) \neq x(e)$ if $x(e) \in A$ whereas $f_e\big(x(e)\big) = x(e)$ if $x(e) \in B$. Hence, f_e is not a nontrivial automorphism of $\mathscr{L}$. $\square$

This concludes the proof of **Theorem 3.30**. $\square$

Corollary 3.36. *For every* Σ^0_2 *set* $\mathcal{A} \subseteq \mathcal{O}$*, there exists a computable linear ordering of order type* $\boldsymbol{\omega} + \boldsymbol{\zeta}$ *which is* $\Sigma^{-1}_{\mathcal{A}}$*-rigid.*

Furthermore, we apply the same argument as in the proof of **Theorem 3.30** to the similar order types.

Corollary 3.37. *For any graph subuniform* Δ^0_2 *class* $\mathcal{F}$*, there exists a computable linear ordering which is* $\mathcal{F}$*-rigid and has one of the following order types*

$$\gamma_0\tau_0\gamma_1\tau_1\cdots\gamma_n\tau_n$$

with $n \geq 0$ *and* γ_i *and* τ_i *being of order type* ω *and* ζ *respectively for all* $i \leq n$*.*

3.5 Open Questions

In this section, we suggest further questions concerning the class of computable order types which are $\mathcal{F}$-rigid, where $\mathcal{F}$ is a graph uniform Δ^0_2 class, by posing the following fundamental problem.

Problem 3.38. *Classify the order types* σ *such that* σ *is* $\mathcal{F}$*-rigid — and consequently,* σ *is* $\Sigma^{-1}_{\mathcal{A}}$*-rigid for any* Σ^0_2 *set* $\mathcal{A} \subseteq \mathcal{O}$*;* σ *is* $\mathcal{G}$*-rigid for the class* $\mathcal{G}$ *of* a*-c.e. functions,* $a \in \mathcal{O}$*; etc.*

There are quite deep questions, and they might possibly lead to progress with **Problem 3.38** and to a more general approach to automorphisms of linear orderings and their constructive character.

Conjecture 3.39. *For any graph subuniform* Δ^0_2 *class* $\mathcal{F}$ *and for the order type* $\boldsymbol{n}$ *of* n*-element chain, there exists a computable linear ordering of order type* $\boldsymbol{n} \cdot \boldsymbol{\eta}$ *which is* $\mathcal{F}$*-rigid.*

Conjecture 3.40. *For any graph subuniform Δ^0_2 class $\mathcal{F}$, there exists a computable linear ordering of order type $\boldsymbol{\zeta} \cdot \boldsymbol{\eta}$ which is $\mathcal{F}$-rigid.*

Bibliography

[1] Marat M. Arslanov. The Ershov hierarchy. In S. Barry Cooper and Andrea Sorbi, editors, *Computability in Context: Computation and Logic in the Real World*. Imperial College Press, 2011.

[2] Christopher J. Ash and Julia F. Knight. *Computable Structures and the Hyperarithmetical Hierarchy*. Elsevier, 2000.

[3] Christopher J. Ash and Anil Nerode. Intrinsically recursive relations. In John N. Crossley, editor, *Aspects of Effective Algebra (Proceedings of a conference, Monash University, Clayton, Australia, August 1-4, 1979)*, pages 26–41. Upside Down A Book Co., 1981.

[4] Robert Bonnet. Stratifications et extension des genres de chaînes dénombrables. *Comptes rendus de l'Académie des Sciences*, 269:880–882, 1969.

[5] Robert Bonnet and Maurice A. Pouzet. Extension et stratification d'ensembles dispersés. *Comptes rendus de l'Académie des Sciences*, 268:1512–1515, 1969.

[6] Robert Bonnet and Maurice A. Pouzet. Linear extensions of ordered sets. In Ivan Rival, editor, *Ordered Sets*, volume 83 of *NATO Advanced Science*

Institutes Series C: Mathematical and Physical Sciences, pages 125–170, 1982.

[7] S. Barry Cooper. *Computability theory.* Chapman & HALL/CRC Mathematics, 2004.

[8] S. Barry Cooper, Charles M. Harris, and Kyung Il Lee. Automorphisms of computable linear orderings and the Ershov hierarchy. In preparation.

[9] S. Barry Cooper, Kyung Il Lee, and Anthony Morphett. Linearisations in the Ershov hierarchy. In preparation.

[10] Rodney G. Downey. Computability theory and linear orderings. In Yuri L. Ershov, Sergei S. Goncharov, Anil Nerode, and Jeffrey B. Remmel, editors, *Handbook of Recursive Mathematics Volume 2: Recursive Algebra, Analysis and Combinatorics*, Studies in Logic and the Foundations of Mathematics, pages 823–976. North-Holland, 1998.

[11] Rodney G. Downey, Denis R. Hirschfeldt, Steffen Lempp, and David R. Solomon. Computability-theoretic and proof-theoretic aspects of partial and linear orderings. *Israel Journal of mathematics*, 138:271–289, 2003.

[12] Rodney G. Downey and Steffen Lempp. The proof-theoretic strength of the Dushnik-Miller theorem for countable linear orders. In *De Gruyter Series in Logic and Its Applications: Recursion Theory and Complexity: Proceedings of the Kazan '97 Workshop*, volume 271, pages 55–58, 1999.

[13] Rodney G. Downey and Michael F. Moses. On choice sets and strongly non-trivial self-embeddings of recursive linear orders. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 35:237–246, 1989.

[14] Ben Dushnik and Edwin W. Miller. Concerning similarity transformations of linearly ordered sets. *Bulletin of the American Mathematical Society*, 46:322–326, 1940.

[15] Yuri L. Ershov. A hierarchy of sets, I. *Algebra i Logika*, 7(1):47–74, January–February 1968. English translation, Consultants Bureau, NY, pp. 25–43.

[16] Yuri L. Ershov. On a hierarchy of sets, II. *Algebra i Logika*, 7(4):15–47, July–August 1968. English translation, Consultants Bureau, NY, pp. 212–232.

[17] Yuri L. Ershov. On a hierarchy of sets, III. *Algebra i Logika*, 9(1):34–51, January–February 1970. English translation, Consultants Bureau, NY, pp. 20–31.

[18] Lawrence Feiner. *Orderings and Boolean algebras not isomorphic to recursive ones.* PhD thesis, Massachusetts Institute of Technology, 1967.

[19] Rechard M. Friedberg. Two recursively enumerable sets of incomparable degrees of unsolvability (solution to post's problem, 1944). *Proceedings of the National Academy of Sciences of the United States of America*, 43:236–238, 1957.

[20] Rechard M. Friedberg. Three theorems on recursive enumeration. I. decomposition. II. maximal set. III. enumeration without duplication. *Journal of Symbolic Logic*, 23:309–316, 1958.

[21] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types.* Cambridge University Press, 1989.

[22] Kurt Gödel. On undecidable propositions of formal mathematical systems. In Martin Davis, editor, *The Undecidable. Basic Papers on Undecidable*

Propositions, Unsolvable Problems and Computable Functions, pages 39–71, New York, 1934. Raven Press, Hewlitt. First appeared in the form of mimeographed notes on lectures given by Gödel at the Institute for Advanced Study during the spring of 1934.

[23] E. Mark Gold. Limiting recursion. *J. Symbolic Logic*, 30(1):28–48, 1965.

[24] V.S. Harizanov. Pure computable model theory. In Yuri L. Ershov, Sergei S. Goncharov, Anil Nerode, and Jeffrey B. Remmel, editors, *Handbook of Recursive Mathematics, Volume 1: Recursive Model Theory*, Studies in Logic and the Foundations of Mathematics, pages 3–114. Elsevier, 1998.

[25] Leo A. Harrington. A gentle approach to priority arguments. 1982. Mimeographed Notes.

[26] Carl G. Jockusch, Jr. Degrees in which the recursive sets are uniformly recursive. *Canadian Journal of Mathematics*, 24:1092–1099, 1972.

[27] Henry A. Kierstead. On Π_1-automorphisms of recursive linear orders. *Journal of Symbolic Logic*, 52:681–688, 1987.

[28] S.C. Kleene. The theory of recursive functions, approaching its centennial. *Bulletin (New Series) of the American Mathematical Society*, 5(1):43–61, 1981.

[29] Stephen C. Kleene. On notation for ordinal numbers. *Journal of Symbolic Logic*, 3(4):150–155, 1938.

[30] Stephen C. Kleene. *Introduction to Metamathematics.* Van Nostrand, Princeton, 1952.

[31] Stephen C. Kleene. On the forms of the predicates in the theory of constructive ordinals (second paper). *American Journal of Mathematics*, 77(3):405–428, 1955.

[32] Stephen C. Kleene. *Formalized recursive functionals and formalized realizability*, volume 89 of *A.M.S Memoirs*. 1969.

[33] Stephen C. Kleene. Recursive functionals and quantifers of finite types revisited i. In Jens E. Fenstad, Robin O. Gandy, and Gerald E. Sacks, editors, *Generalized Recursion Theory II*, pages 185–222. North-Holland, 1978.

[34] Stephen C. Kleene and Emil L. Post. The upper semi-lattice of degrees of recursive unsolvability. *Annals of Mathematics (2)*, 59:379–407, 1954.

[35] Alistair H. Lachlan. The priority method for the construction of recursively enumerable sets. In A. R. D. Mathias and Hartley Rogers, Jr., editors, *Cambridge Summer School in Mathematical Logic*, volume 337 of *Lecture Notes in Mathematics*, pages 299–310, Berlin, 1973. Springer–Verlag.

[36] Alistair H. Lachlan. A recursively enumerable degree which will not split over all lesser ones. *Annals of Mathematical Logic*, 9:307–365, 1975.

[37] Manuel Lerman and Joseph G. Rosenstein. Recursive linear orderings. In *Logic Symposium at Patras 1980*, pages 123–126, 1981. Math Reviews 84j:03092.

[38] Manuel Lerman and James H. Schmerl. Theories with recursive models. *Journal of Symbolic Logic*, 44:59–76, 1979.

[39] W. Markwald. Zur Theorie dler konstruktiven Wollordcnunge. *Mathernatische Annalen*, 127:135–149, 1954.

[40] Yu. T. Medevdev. Degrees of difficulty of the mass problems. *Doklady Akademii Nauk SSSR*, 104:501–504, 1955.

[41] Michael F. Moses. *Recursive properties of isomophism types.* PhD thesis, Monash University, Clayton, Victoria, Australia, 1983.

[42] Michael F. Moses. Recursive linear orders with recursive successivities. *Annals of Pure and Applied Logic*, 27:253–264, 1984.

[43] Albert A. Mučnik. On the unsolvability of the problem of reducibility in the theory of algorithms. *Doklady Akademii Nauk SSSR*, N. S. 108:194–197, 1956.

[44] André Nies. *Computability and Randomness.* Oxford Logic Guides. Oxford University Press, 2009.

[45] Piergiorgio G. Odifreddi. *Classical Recursion Theory, Volume I.* North-Holland, Amsterdam, 1989.

[46] Piergiorgio G. Odifreddi. *Classical Recursion Theory, Volume II.* North-Holland, Amsterdam, 1999.

[47] Mikhail G. Peretyat'kin. Every recursively enumerable extension of a theory of linear order has a constructive model. *Algebra i Logika*, 12:211–219, 244, 1973. English translation, Algebra and Logic, Vol. 12, pp. 120–124, 1973.

[48] Emil L. Post. Recursively enumerable sets of positive integers and their decision problems. *Bulletin of the American Mathematical Society*, 50:284–316, 1944.

[49] Hillary Putnam. Trial and error predicates and the solution to a problem of Mostowski. *Journal of Symbolic Logic*, 30(1):49–57, 1965.

[50] Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York, 1967.

[51] Joseph G. Rosenstein. *Linear Orderings.* Academic Press, 1982.

[52] Joseph G. Rosenstein. Recursive linear orderings. In *Orders: description and roles*, pages 465–475, 1984.

[53] Gerald Sacks. *Higher Recursion Theory.* Perspectives in Mathematical Logic. Springer–Verlag, Heidelberg, 1990.

[54] Leonard P. Sasso, Jr. *Degrees of unsolvability of partial functions.* PhD thesis, University of California at Berkeley, 1971.

[55] Steven Schwarz. Recursive automorphisms of recursive linear orderings. *Annals of Pure and Applied Logic*, 25:69–73, 1984.

[56] Joseph R. Shoenfield. On degrees of unsolvability. *Annals of Mathematics (2)*, 69:644–653, 1959.

[57] Richard A. Shore. A non-inversion theorem for the jump operator. *Annals of Pure and Applied Logic*, 40:277–303, 1988.

[58] Stephen G. Simpson. *Subsystems of second order arithmetic.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1999.

[59] Robert I. Soare. Tree arguments in recursion theory and the $0'''$-priority method. In Anil Nerode and Richard A. Shore, editors, *Recursion Theory: Proceedings of Symposia in Pure Mathematics*, volume 42, pages 53–106, 1985.

[60] Robert I. Soare. *Recursively Enumerable Sets and Degrees.* Perspectives in Mathematical Logic, Ω Series. Springer-Verlag, Heidelberg, 1987.

[61] Robert I. Soare. Computability and recursion. *Bulletin of Symbolic Logic*, 2(3):284–321, 1996.

[62] Robert I. Soare. Computability and incomputability. In *CiE 2007: Computation and Logic in the Real World*, pages 705–715. Springer, 2007.

[63] Robert I. Soare. *Draft 2008 - Computability theory and applications*. 2008.

[64] Robert I. Soare. *Draft 2009 - Computability theory and applications*. 2009.

[65] Robert I. Soare. Turing oracle machines, online computing, and three displacements in computability theory. *Annals of Pure and Applied Logic*, 160(3):368–399, 2009.

[66] Robert I. Soare. Turing and the art of classical computability. In S. Barry Cooper and Jan van Leuwan, editors, *Alan Turing His Work and Impact*. Elsevier, 2012.

[67] Frank Stephan, Yue Yang, and Liang Yu. Turing degrees and the Ershov hierarchy. Technical report, National University of Singapore, 2009.

[68] Edward Szpilrajn. Sur lextension de lordre partiel. *Fundmenta Mathematica*, 16:386–389, 1930.

[69] Alan M. Turing. On computable numbers with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265, 1936. A correction, 43:544–546.

[70] Alan M. Turing. Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, 45:161–228, 1939. Reprinted in Alan M. Turing, Collected Works: Mathematical Logic, pp. 81–148.

[71] Richard M. Watnick. A generalization of Tennenbaum's theorem on effectively finite recursive linear orderings. *Journal of Symbolic Logic*, 49(2):563–569, 1984.